

---

StatPac for Windows

# User's Guide

By StatPac Inc.



***Survey Software  
for  
Professional Researchers***

StatPac for Windows software and manual

Copyright © 1999-2015 by StatPac Inc.

All Rights Reserved.

ISBN 0-918733-12-X

StatPac is a trademark of StatPac Inc.

StatPac Inc.  
1200 First Street  
Pepin, Wisconsin 54759  
Tel: (715) 442-2261  
Fax: (715) 442-2262

# Contents

<b>Introduction</b>	<b>1</b>
Overview .....	1
System Requirements and Installation .....	2
Local PC Hardware & Software Requirements .....	2
Server Hardware & Software Requirements .....	2
Other Useful Software .....	3
Installation on a Local PC .....	3
Unregistering & Removing the Software from a PC .....	4
Network Operation .....	4
Updating to a More Recent Version .....	5
Backing-Up a Study .....	6
Processing Time .....	7
Server Demands and Security .....	7
Technical Support .....	8
Notice of Liability .....	8
Paper & Pencil and CATI Survey Process .....	9
Internet Survey Process .....	9
Basic File Types .....	10
Codebooks (.cod) .....	10
Data Manager Forms (.frm) .....	10
Data Files (.dat) .....	10
Internet Response Files (.asc or .txt) .....	10
Email Address Lists (.lst or .txt) .....	11
Email Logs (.log) .....	11
Rich Text Files (.rtf) .....	11
HTML Files (.htm) .....	11
Perl Script (.pl) .....	11
Password Files (.text) .....	12
Exported Data Files (.txt and .csv and .mdb) .....	12
Email Body Files (.txt or .htm) .....	12
Sample File Naming Scheme for a Survey .....	12
Customizing the Package .....	13
 <b>Basic Research Concepts</b>	 <b>15</b>
Problem Recognition and Definition .....	15
Creating the Research Design .....	16
Methods of Research .....	16
Sampling .....	16
Data Collection .....	18
Reporting the Results .....	18
Validity .....	18
Reliability .....	19
Systematic and Random Error .....	20
Formulating Hypotheses from Research Questions .....	20

Type I and Type II Errors .....	21
Types of Data.....	21
Significance .....	24
One-Tailed and Two-Tailed Tests .....	24
Procedure for Significance Testing.....	25
Bonferroni's Theorem .....	26
Central Tendency .....	27
Variability .....	27
Standard Error of the Mean.....	28
Inferences with Small Sample Sizes .....	28
Degrees of Freedom.....	28

## **Codebook Design 29**

Components of a Study Design.....	29
Elements of a Variable.....	30
Variable Format.....	30
Variable Name .....	31
Variable Label .....	32
Value Labels .....	32
Valid Codes .....	33
Skip Codes for Branching.....	34
Data Entry Control Parameters .....	35
Missing OK .....	35
Auto Advance.....	35
Caps Only .....	35
Codebook Tools.....	36
The Grid.....	36
Codebook Libraries .....	37
Duplicating Variables .....	38
Insert & Delete Variables .....	38
Move Variables .....	38
Starting Columns .....	38
Print a Codebook .....	39
Spell Check a Codebook.....	39
Variable Detail Window.....	40
Codebook Creation Process .....	41
Method 1 - Create a Codebook from Scratch .....	41
Method 2 – Create a Codebook from a Word-Processed Document.....	42
Multiple Response Variables .....	42
Missing Data.....	45
Changing Information in a Codebook.....	46

## **Data Manager Form 49**

Overview.....	49
Data Input Fields.....	49
Form Naming Conventions.....	49
Form Creation Process.....	50
Using the Codebook to Create a Form.....	50
Using a Word-Processed Document to Create a Form .....	51
Variable Text Formatting.....	52
Field Placement .....	52
Value Labels .....	53
Variable Separation .....	53
Variable Label Indent .....	53

Value Labels Indent.....	54
Space between Columns.....	54
Valid Codes .....	55
Skip Codes.....	55
Variable Numbers.....	55
Variable List and Detail Windows.....	55
Data Input Settings .....	56
Select a Specific Variable .....	57
Finding Text in the Form .....	58
Replacing Text in the Form .....	58
Saving the Codebook or Workspace.....	59

## **Data Manager 61**

Overview .....	61
Keyboard and Mouse Functions .....	62
Create a New Data File.....	62
Edit or Add To an Existing Data File .....	62
Select a Different Data File .....	62
Change Fields .....	62
Change Records .....	63
Enter a New Data Record .....	63
View Data for a Specified Record Number .....	64
Find Records That Contain Specified Data.....	64
Field To Search .....	65
Search For .....	65
Search Direction .....	65
Search Method.....	65
Duplicate a Field from the Previous Record.....	65
Delete a Record.....	66
Data Input Settings .....	66
Compact Data File .....	67
Double Entry Verification .....	68
Print a Data Record.....	68
Variable List & Detail Windows .....	68
Data File Format.....	68

## **Email Surveys 71**

Overview .....	71
HTML Email Surveys.....	71
Plain Text Email Surveys .....	72
Brackets.....	73
Item Numbering .....	73
Codebook Design for a Plain Text Email Survey.....	74
Capturing a Respondent's Email Address.....	76
Filtering Email to a Mailbox .....	76
General Considerations for Plain Text Email .....	76

## **Internet Surveys 77**

Overview .....	77
Internet Survey Process .....	77
Server Setup .....	77
Create the HTML Survey Pages.....	78
Upload the Files to the Web server .....	78
Test the survey.....	78

Download and import the test data .....	78
Delete the test data from the server .....	78
Conduct the survey .....	78
Download and import the data.....	78
Display a survey closed message.....	79
Server Setup.....	79
FTP Login Information.....	80
Paths & Folder Information.....	80
Design Considerations for Internet Surveys .....	84
Special Variables for Internet Surveys.....	85
Script to Create the HTML .....	86
Command Syntax & Help.....	88
Saving and Loading Styles.....	89
Survey Generation Procedure .....	89
Script Editor.....	90
Imbedded HTML Tags .....	90
Primary Settings.....	91
HTML Name .....	91
Banner Image.....	93
Heading .....	93
Finish Text & Finish URL.....	93
Cookie .....	93
IP Address Control .....	94
Allow Cross Site Access.....	94
URL to Survey Folder .....	95
Advanced Settings .....	96
Header & Footer .....	96
Finish & Popups .....	99
Control.....	101
Fonts & Colors .....	104
Passwords - Color & Banner Image .....	110
Passwords - Text & Control .....	111
Passwords - Single vs. Multiple.....	113
Passwords - Technical Notes .....	115
Server Overrides.....	116
Branching and Piping.....	118
Randomization (Rotations) .....	119
Survey Creation Script.....	120
Overview .....	120
Specify Text.....	121
Spacing and pagination.....	122
Images and Links.....	123
Help Windows .....	124
Popup Windows.....	126
Survey Creation - Objects .....	127
Radio Buttons for a Single Variable .....	127
Radio Buttons for Grouped Variables (matrix style).....	129
DropDown Menu .....	132
TextBox for a Single Variable .....	133
Adding a TextBox to a Radio Button, CheckBox, or Radio Button Matrix .....	136
TextBoxes for Grouped Variables .....	142
Slider for Single or Multiple Variables.....	144
CheckBox for Multiple Response Variables.....	147
Checkbox for Groups of Multiple Response Variables (horizontal matrix) .....	147
ListBox .....	150
Uploading and Downloading Files from the Server.....	150

Auto Transfer .....	150
FTP .....	151
Summary of the Most Common Script Commands .....	155

## **Email List Management 156**

Overview .....	156
Format of an Email Address File .....	156
Extract Email Addresses .....	158
List Statistics .....	158
Join Two or More Lists .....	159
Split a List .....	160
Clean, Sort, and Eliminate Duplicates .....	162
Add ID Numbers to a List .....	162
Create a List of Nonresponders .....	163
Subtract One List From Another List .....	164
Merge an Email List into a StatPac Data File .....	165
Send Email Invitations .....	165
Using an ID Number to Track Responses .....	165
Email Address File .....	166
Body Text File .....	166
Sending Email .....	167

## **Procedure Files 171**

Overview .....	171
Mouse and Keyboard Functions .....	172
Designing Analyses .....	172
Continuation Lines .....	175
Comment Lines .....	175
V Numbers .....	176
Keywords .....	176
Analyses .....	176
Variable List .....	177
Variable Detail .....	177
Find Text or Variable .....	178
Replace Text .....	179
Options .....	179
Load, Save, and Merge Procedure Files .....	184
Print a Procedure File .....	184
Run a Procedure File .....	184
Results Editor .....	186
Graphics .....	186
Table of Contents .....	189
Automatically Generated Topline Procedures .....	190

## **Keywords 191**

Keyword Index .....	191
Keywords Overview .....	191
Categories of Keywords .....	192
Keyword Help .....	192
Ordering Keywords .....	193
Global and Temporary Keywords .....	194
Permanently Change a Codebook and Data File .....	194
Backup a Study .....	195
STUDY Command .....	196

DATA Command.....	197
SAVE Command .....	197
WRITE Command .....	198
MERGE Command.....	202
HEADING Command.....	202
TITLE Command.....	203
FOOTNOTE Command.....	204
LABELS Command.....	204
OPTIONS Command.....	206
SELECT and REJECT Commands.....	208
NEW Command.....	213
LET Command .....	215
STACK Command.....	216
RECODE Command.....	218
COMPUTE Command.....	220
AVERAGE, COUNT and SUM Commands .....	224
IF-THEN ... ELSE Command.....	226
SORT Command.....	232
WEIGHT Command .....	233
NORMALIZE Command .....	235
LAG Command.....	236
DIFFERENCE Command.....	237
DUMMY Command .....	239
RUN Command .....	241
REM Command .....	242
Reserved Words.....	243
Reserved Word RECORD .....	244
Reserved Word TOTAL .....	245
Reserved Word MEAN.....	245
Reserved Word TIME.....	245

## **Basic Analyses 247**

Analyses Index.....	247
Analyses Overview .....	247
LIST Command .....	248
FREQUENCIES Command.....	252
CROSSTABS Command.....	274
BANNERS Command .....	281
DESCRIPTIVE Command .....	299
BREAKDOWN Command.....	305
TTEST Command.....	309
T-Test For Matched Pairs .....	309
T-Test For Independent Groups.....	310
CORRELATE Command .....	313

## **Advanced Analyses 317**

Advanced Analyses Index.....	317
REGRESS Command .....	317
STEPWISE Command.....	325
LOGIT and PROBIT Commands .....	336
PCA Command.....	342
FACTOR Command .....	349
CLUSTER Command .....	355
DISCRIMINANT Command.....	364



ANOVA Command .....	373
CANONICAL Command .....	393
MAP Command .....	400
Advanced Analyses Bibliography.....	408

## Utilities 411

Utility Programs.....	411
Import and Export.....	412
StatPac and Prior Versions of StatPac Gold .....	413
Access, Excel, Paradox, dBase, and Lotus .....	413
DIF Files.....	414
Comma Delimited and Tab Delimited Files .....	414
Files Containing Multiple Data Records per Case.....	415
Internet Files.....	416
Excel Files .....	416
Email Surveys.....	416
Merging Data Files .....	417
Concatenate Data Files .....	418
Merge Variables and Data .....	418
Aggregate.....	420
Codebook.....	423
Quick Codebook Creation .....	423
Check Codebook and Data .....	425
Sampling.....	425
Random Number Table .....	426
Random Digit Dialing Table .....	427
Select Random Records from Data File .....	428
Create Variable for Weighting .....	429
Compare Data Files .....	432
Conversions .....	432
Date Conversions .....	433
Currency Conversion.....	434
Dichotomous Multiple Response Conversion .....	435

## Statistics Calculator 439

Statistics Calculator Menu .....	439
Distributions Menu .....	440
Normal distribution .....	440
T distribution .....	441
F distribution .....	443
Chi-square distribution .....	443
Counts Menu.....	444
Chi-square test.....	444
Fisher's Exact Test.....	445
Binomial Test .....	446
Poisson Distribution Events Test.....	446
Percents Menu .....	446
Choosing the Proper Test .....	446
One Sample t-Test between Percents .....	448
Confidence Intervals around a Percent.....	450
Means Menu .....	451
Mean and Standard Deviation of a Sample .....	451
Matched Pairs t-Test between Means .....	452
Independent Groups t-Test between Means .....	453

Confidence Interval around a Mean.....	454
Compare a Sample Mean to a Population Mean.....	455
Compare Two Standard Deviations.....	456
Compare Three or more Means .....	456
Correlation Menu .....	458
Regression .....	459
Sampling Menu.....	461
Sample Size for Percents .....	461
Sample Size for Means .....	462

## **Index**

**465**

# Introduction

---

## Overview

StatPac is a complete data manager and analysis package. It will handle all phases of survey design and analysis.

The steps in running StatPac are the same as in all research:

1. Design the study
2. Collect the data
3. Run the analyses

Numerous other tasks may be performed, such as managing e-mail lists and manually entering/editing data.

The study design contains a description of the variables, their labels, and the data file format. The study design is stored in a file called the **codebook**. All codebook file names end with a .cod extension.

Another part of the study design is called a **form**. The form is used for data entry and editing. You do not need a form unless you will be doing manual data entry or editing. All form names end with a .frm extension. When you are processing questionnaires, the form closely resembles the questionnaire itself. StatPac's data manager uses the form to allow entry and editing of data.

An essential element of StatPac is the ability to create and maintain a database of information to be analyzed. This may be questionnaire data, test scores or any other type of "raw" information. The information is stored in a **data file** on disk. All data file names end with a .dat extension.

When performing a Web survey, the responses will be stored on the server in an ASCII text file using a .asc extension. (Note: The default extension for internet response files may be changed by modifying the InternetExtension setting in the StatPac.ini file.) When you're ready to perform analyses, you'll download the **response file** to your local computer and import it into a StatPac data file.

The final step is to perform the analyses. StatPac is designed for either interactive or batch processing. This means you can run a single analysis (interactive) or many different analyses at one time (batch). To run an analysis you will type a set of commands that say "first do this", "next do this", and so on. These commands are stored in a **procedure file**. Procedure file names end with a .pro extension.

---

# System Requirements and Installation

Installing StatPac on a hard disk is very easy and will take about five minutes.

If you have any problems installing this product, please do not hesitate to write, call, or e-mail.

StatPac Inc.  
Technical Support  
1200 First Street  
Pepin, WI 54759

(715) 442-2261 (9-5 Central Time)  
(715) 442-2262 (Fax)  
support@statpac.com

Please note that StatPac can legally be installed on two computers provided that there will not be simultaneous use of both installations. It may be installed on a network drive. However, access to the software will be restricted to the workstation used for the installation. Any other installations are a violation of copyright laws. If you wish to install StatPac on an additional computer, please remove it from the current computer before installing it on the new computer, or purchase a second copy at a reduced cost.

When conducting web surveys, the HTML files that StatPac creates may be installed on any server or servers. No special license is required to upload the HTML files to multiple servers. Additionally, the Perl scripts provided with the software may be installed on more than one server. No special license is required for multiple installations of the Perl scripts.

## Local PC Hardware & Software Requirements

StatPac will work on a PC running Windows 95, Windows 98, Windows 2000, NT, XP, Vista, Windows 7, 8 or 10. It requires about 50Mb of disk space and a minimum of 500M RAM. StatPac's performance is directly related to the CPU clock speed (faster is better) and the amount of RAM (more is better). We suggest not using StatPac on older PCs that have a CPU clock speed slower than 300MHz.

## Server Hardware & Software Requirements

If you will be conducting Web surveys, you'll need access to a Web server to "host" your surveys. The server may be Unix/Linux or any version of Windows/IIS. It must support CGI. This means you will have access to a cgi-bin folder on your server. Nearly all hosting services support CGI, so you may need to contact your ISP for more information. StatPac has a Perl script that you will be installed in the cgi-bin folder on your server. If you do not currently have your own Web server, you may use StatPac's free server. The domain for our hosting server is take-survey.com.

## Other Useful Software

All StatPac reports are created in rich text format. These files can be viewed, printed, and manipulated in StatPac, or you can use any word processor to view the reports (e.g. Microsoft Word).

For Web surveys, you may also wish to use a WYSIWYG HTML editor. StatPac will create aesthetically pleasing and fully functional Internet surveys, but you may want to visually enhance their appearance with additional graphics or other design features. In order to do that, you must have a **What-You-See-Is-What-You-Get** HTML editor. Microsoft Front Page, Front Page Express, and DreamWeaver are examples of a WYSIWYG HTML editor. Any WYSIWYG HTML editor will work. While we do not recommend it, you may also use recent versions Microsoft Word to edit your HTML files.

## Installation on a Local PC

1. Choose **Run** from the Start Bar.
2. Use the Browse button to navigate to your CD (or the file you downloaded from our Web site). Click on "Install StatPac" (on Windows 7 and higher, right click and select Run As Administrator)..
3. Click OK to install the package. The default installation folder is C:\StatPac.

System administrators: Users must have read/write access to the folder where the software is installed. Since the Vista, Windows 7, 8, and 10 operating systems do not allow users write access to the Program Files folder, it should not be installed in the Programs Files folder. We suggest leaving the installation folder as C:\StatPac.

4. After the installation has completed, the Microsoft Help File Reader must be installed on Vista, Windows 7, 8, and 10 machines. Use Windows Explorer to navigate to the C:\StatPac folder. Install the appropriate Microsoft help file reader by clicking on it. If you do not know whether you have a 32 bit or 64 bit machine, try first installing the 64 bit version. It will tell you if it did not install correctly and then you can install the 32 bit version. The Help File Reader files are named:

Vista-Help-32.msu  
Vista-Help-64.msu  
Windows7-Help-32.msu  
Windows7-Help-64.msu  
Windows8-Help-32.msu  
Windows8-Help-64.msu

5. The StatPac installation will create an icon on your desktop.



Double click on the icon to run StatPac

6. Select Help, Enter Unlock Code
7. Type your User Name, Serial Number and UNLOCK Code and click OK.
8. Answer 'Yes' to connect to the StatPac server. This will turn the demo into a full version. If you do not have an internet connection or if you are unable to connect to the StatPac server, press [Esc] instead of answering 'Yes'. Then call or email StatPac Inc. for an authorization key. Some firewalls block StatPac's online registration. Your network supervisor can adjust your firewall to allow your PC to communicate with our server. The IP address of the StatPac registration server is 74.220.220.50

Note: If you plan to use our server to host your online surveys, then your network supervisor should also unblock IP address: 74.220.201.161 which is the address for our hosting server take-survey.com. StatPac uses standard FTP protocol, which means that ports 20, 21, and all ports above 1023 must be unblocked for that IP address.

That completes the installation on your local computer.

---

## Unregistering & Removing the Software from a PC

StatPac for Windows is copyrighted and should not be registered on more than one computer at a time (without a Network License Agreement) with the following single exception. StatPac Inc. specifically grants you the right to install the software on one additional computer, provided that both computers will not run the software simultaneously.

Examples of typical installations would be: 1) one home and one office computer, 2) one office computer and a laptop computer, 3) one main computer and one backup computer, 4) your computer and your technical support person's computer. You are specifically prohibited from installing two copies of the software where there will be two simultaneous users of the software.

When you register the software, it is converted from a demo version to a full version. After registering StatPac, you can unregister it from one machine and then register it on another machine. When you unregister the software, it is converted from a full version back to a demo.

Unregistering the software will enable you to install and register the package on a different computer. Select Help, Enter UNLOCK Code. Type [Ctrl S]. A minus sign will be added to the beginning of the serial number. Type your UNLOCK code. Click OK to unregister the software. The copy on that computer will be changed to a demo version and you will then be able to register it on another computer. It is not necessary to actually remove the package after unregistering it. You may keep the demo version on that computer so it will be easy to reregister it on that computer in the future.

After unregistering the package, you may remove it from that computer by selecting Control Panel, Add/Remove Programs, and click on StatPac All the files from the directory where you installed the software will be removed. All files that you created (codebooks, data files, etc.) will not be deleted.

---

## Network Operation

When you purchase a Network License Agreement, your serial number and password will automatically activate the network option. The network option will not work unless you have a special serial number and password.

The software must be installed from each station that will have access to the network. At the first station, install the software to a network drive. Subsequent installations at different stations should install to the same network drive and folder. Some required files will be written to the local computer's Windows\System folder. Use your serial number and UNLOCK code to activate each station after you install it.

After installing the software on all stations, perform the following two steps to use the networking capabilities of StatPac for Windows.

1. Create or decide upon a folder where user profiles will be stored. The folder can have any name. All users must have permission to write to this folder. A profile is the same as the StatPac.ini file and contains all the default values for the software. Each user will have their own profile (their own default values).

For example, you might create a folder called:

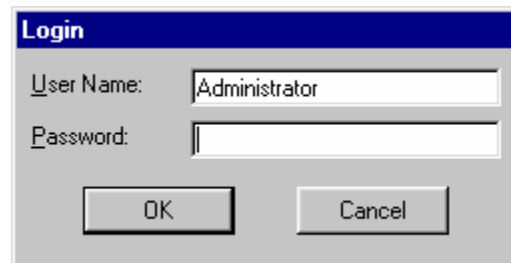
**C:\StatPac\UsersFolder**

2. Create an ASCII text file (using notepad or any word processor) and save it in the StatPac programs folder using the name "Users.ini". The first line of the text is the path to the folder where user profiles are stored. This may be the fully qualified path, or can be a path relative to the folder where StatPac was installed. Subsequent lines in the text file are user names and passwords separated by commas. Do not include any spaces unless they are actually part of the password. Upper and lower case characters are different.

For example, three user names and passwords are specified in this "Users.ini" file:

**C:\StatPac\UsersFolder  
Administrator,Boss  
David,programmer  
Hilda,Pastor**

StatPac networking will now be enabled. User profiles will automatically be created and stored in the user profile folder when the user first logs into StatPac. The login screen will appear each time a user runs the package.



If you need to add a new user, first temporarily rename the Users.ini file to something else. Then install the software from the new station and activate it with the serial number and password. Finally, rename the file back to Users.ini and edit the file to include the new username and password.

---

## Updating to a More Recent Version

We frequently update the software with enhancements and bug fixes. Please check our web site to see if you have the most recent update. Updates are available to all users who have a current technical support agreement.

**How to update your StatPac:**

1. Run StatPac
2. Select Help, Software Updates
3. Click the Check for Updates button
4. Download the updates
5. Close StatPac and restart it.

Alternatively, set your browser to:

<https://statpac.com/updates/login.htm>

---

## Backing-Up a Study

System crashes are not common; however, when they do happen, it can be devastating. Making backup copies of your study files is an important part of any data analysis procedure. Generally, you should make a backup whenever you feel "it's more than you'd care to lose".

**AT A MINIMUM, ALWAYS BACK UP A STUDY BEFORE BEGINNING ANY ANALYSES OF THAT STUDY.**

The analysis portion of StatPac is very powerful. Variables and data may be easily changed. Furthermore, these changes can easily be made a permanent part of the study. If you should make an error, your study information and data will reflect this error. The only way to undo an error (i.e., to restore the codebook and data to its former state) is to use a back up. **IF YOU HAVE NOT MADE A BACKUP, IT WILL NOT BE POSSIBLE TO UNDO PERMANENT CHANGES YOU HAVE MADE TO THE CODEBOOK OR DATA.**

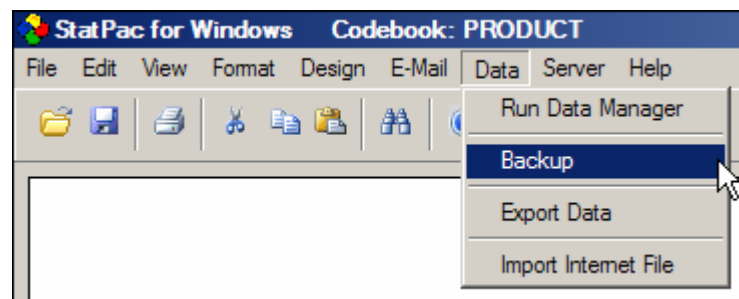
We strongly recommend that you make frequent backup copies of all StatPac codebook and data files.

There are three ways to back up your work:

1) Use Windows Explorer to copy the codebook (.cod), data entry form (.frm), data file (.dat) and procedure file (.pro) to a different folder.

2) Load the codebook for the study and then select Data, Backup. You'll then be able to create a backup folder and all the files associated with the study will be copied to the backup folder. The new backup folder will be created in the same folder as the current codebook.

Note: In most situations, it is desirable (but not necessary) to create a separate folder for each study. Storing each study in its own folder will make it easier to backup and manage the files associated with each study.





3) Begin each procedure file with a simple procedure that writes a duplicate codebook and data file. Then perform subsequent procedures on the duplicate files rather than the originals. By doing this, you'll be leaving the original codebook and data files intact. No matter what happens, you'll always be able to revert back to your original files.

In this example, the original codebook and data file are called MASTER. The first procedure (first 2 lines) writes a new codebook and data file called TEMP. The second procedure begins using the TEMP codebook and data file, and all subsequent procedures will use the TEMP codebook and data file. If you make an erroneous transformation to the data in a subsequent procedure, you'll be able to revert back to the original codebook and data file by re-running the first procedure.

```
STUDY MASTER
WRITE TEMP
..
STUDY TEMP
(all the rest of the procedures)
```

---

## Processing Time

The execution time for any analysis depends upon (in the order of importance):

1. The clock speed of the CPU in your computer.
2. The number of records in the data file.
3. The number and type of transformations being performed.
4. The amount of RAM in the computer.
5. The type of analysis being performed.
6. The options used on the analysis.

When you have several analyses to perform, batch processing can save time. The advantage is that you can submit many analyses at one time and they all get added to the same document.

---

## Server Demands and Security

When conducting Web surveys, the demands on your server will be minimal. Generally, the HTML files created will be small and load quickly into a browser. The server is only called upon when a survey page is being loaded into the respondent's browser and when they click the next page or submit button. Once a survey page has been loaded into the respondent's browser, no demands are placed on the server until the respondent submits that page.

The response data is collected on the server and usually stored in the cgi-bin folder. This folder is protected from unauthorized users. Thus, the information you collect is as secure as your cgi-bin folder.

Additionally, you may add password protection to a survey to only allow access to people who have the password.

---

## Technical Support

Most of the features in StatPac have been developed as a result of user suggestions. If you would like to suggest a program enhancement, please feel free to write us about it. All users of StatPac will benefit.

### **Technical support agreement:**

Demo users receive 30 days of free technical support.

Purchasers of the software receive 90 days of free technical support. After that, you can purchase a technical support agreement. To renew your technical support agreement, select Help, Technical Support, and click on the link to renew your support agreement.

The support agreement includes up to 90 minutes of support for a period of one year. It also includes updates for that year. The cost for updates and a support agreement depends upon how long you have gone without an agreement.

### **How to find the serial number of your package:**

Click Help, About StatPac for Windows to see the serial number of your software.

Demo versions do not have a serial number. Thirty days of free support is provided to demo users.

### **Tech support by e-mail:**

1. Include the serial number of your package.
2. Include your codebook (.cod), data file (.dat) and procedure file (.pro) as attachments. This will enable us to reproduce the problem. A zipped file is preferred.
3. Email to: support@statpac.com

### **Tech support by telephone:**

1. The serial number of your package. Click Help, About StatPac for Windows to see your serial number. (Demo versions do not have a serial number.)
2. All relevant printouts in front of you.
3. Please try to be at the computer and phone at the same time.
4. Call (715) 442-2261 between 9:00 AM and 5:00 PM Central time.

---

## Notice of Liability

StatPac for Windows is distributed on an "AS IS" basis without warranty.

StatPac Inc. has made extensive tests on StatPac for Windows and believes it to be "bug" free. However, in any set of programs as complex as StatPac for Windows, there is the possibility that the programs will malfunction with unusual or invalid data.

StatPac Inc. shall have no liability or responsibility to customer or any other person or entity with respect to any liability, loss or damage caused or alleged to be caused directly or indirectly by StatPac for Windows. This includes, but is not limited to, any interruption of service, loss of data, loss of business or anticipatory profits, or consequential damages from the use of StatPac for Windows.

Information in this document is subject to change without notice and does not represent a commitment on the part of StatPac Inc. The software described in this document is furnished under a license agreement. The software may be used or copied only in accordance with the terms of the agreement.

---

## Paper & Pencil and CATI Survey Process

Paper and pencil surveys and CATI (computer assisted telephone interviewing) surveys follow the same process:

1. Create a codebook (i.e., the study design). The codebook contains the questions and response choices for all the items on the survey. It also contains validity checking information and branching information that allows the survey to skip to other questions depending on a respondent's answer.
2. Create a data entry/editing form. The data entry form is a template that can be used for entering and editing data. StatPac can automatically generate a nicely formatted form. Once created, forms can be easily modified (such as adding special instructions to interviewers or data entry operators).
3. Enter the raw data. The data manager is used to enter the respondents' answers into a database. Internally, StatPac stores the data in a fixed record-length sequential ASCII file with a carriage return and line feed at the end of each data record.
4. Create a procedure file and run the reports. A procedure file is simply a set of instructions that tells StatPac what kind of reports you want.

---

## Internet Survey Process

Building an Internet survey can be broken down into several distinct steps. The basic process for creating Internet surveys with an e-mail invitation and follow-up reminder to non-respondents is as follows:

1. Create a codebook. The codebook for Internet surveys is nearly identical to the codebook for paper and pencil surveys.
2. Create a default script by selecting Design, Internet Survey. The script controls how the HTML survey pages will be created.
3. Modify the default script as necessary.
4. Generate and view the HTML files. Repeat steps 3 and 4 as necessary.
5. Upload the files to the Web site and test the survey online. This means that you complete the survey as if you were a respondent. Then download the captured data and look at it to make sure it is correct.
6. Create the email invitation body and a test email list containing your email. Send the test email to the test list. When you receive the test email, try the link and make sure it works as expected.
7. Clean the real email list (and add ID numbers to the list if you intend to track who responded). Send e-mail invitations to participate in the survey.

8. Download the file containing visitor responses and capture them into a data file.
9. Create an e-mail list of nonresponders. Send a reminder e-mail to the nonresponders.
10. Download the file containing visitor responses and capture them into a data file.
11. Create a procedure file and run the reports in StatPac or export the data to Access, or a tab or comma delimited file.

---

## Basic File Types

StatPac creates and uses several types of files. The file extensions are controlled by the software and usually may not be changed.

### Codebooks (.cod)

The survey design is stored in a file called the **codebook**. Each item on a survey is a variable. The survey design contains a description of the variables, their labels, and their formats. Additional information, such as skip patterns (branching) and validity checking are also stored in the codebook. All codebook file names end with a .cod extension.

### Data Manager Forms (.frm)

Another part of the study design is called a form. The form is used as a screen template for manually entering and editing. It is not uncommon for a small number of respondents to want to complete a hardcopy of the survey rather than the Internet version. In those situations, you will have to manually add their information to the rest of the data. All forms end with a .frm extension.

### Data Files (.dat)

Data files can be created with the data manager, imported from a file created by another program, or captured from an e-mail or Internet response file.

When conducting a web survey, respondents' answers are stored in a response file on your server. These response files must be downloaded and captured before they can be used or exported. When you capture a response file from the server, it is converted to a fixed-format sequential ASCII file. Responses from multiple page surveys are combined into a single data record per respondent. Data file names end with a .dat extension.

### Internet Response Files (.asc or .txt)

When a respondent completes a survey on your web site, their answers are stored on your server in an ASCII text file. This file is not directly useable by other software; it must first be captured by StatPac and converted into a fixed-format ASCII text file. StatPac supports multiple page surveys by storing the responses to each page as it is completed. If a respondent abandons the survey before completing all pages, you will still have captured the information from the pages that they finished. Internet response file names normally end with an .asc extension, although you can alternatively use a .txt extension. The InternetExtension setting in the StatPac.ini defaults file can be used to set the Internet response file extension.

StatPac provides two methods for uploading and downloading files from your server (Server, Auto Transfer and Server, FTP). If you use your own FTP client to upload and download files, then you must tell your FTP program that files with a .asc extension are to be treated as ASCII (not binary) files. If you are unable to make that setting in your FTP program, then you should change StatPac's default extension for internet response files from .asc to .txt. The InternetExtension setting in the StatPac.ini file sets the extension.

## Email Address Lists (.lst or .txt)

Many web surveys use e-mail invitations to request participation from respondents. Email addresses are stored in an ASCII text file with one e-mail address per line. All mailing list file names created by StatPac end with a .lst or .txt extension. You can specify which extension you want to use by changing the ListExtension setting in the StatPac.ini defaults file. When an email address file contains more than just the email addresses (e.g., ID numbers, demographic information, etc.), fields can be separated from each other by tabs or commas.

## Email Logs (.log)

The StatPac web survey component has a bulk e-mailing program that lets you send thousands of individualized e-mail invitations in one batch. Each time you send to an e-mail list, a log is kept of the successful and unsuccessful e-mails. The log file will have the same file name as the e-mail list except the extension is .log. Each time you send to a particular e-mail list, the log for that list will be appended.

## Rich Text Files (.rtf)

Many people initially develop their surveys using a word processor. When saved in rich text format, these documents can be used to facilitate the design of a codebook. You will be able to copy and paste information from the rich text file to the codebook. Depending on the length of the survey, this may save considerable typing. Rich text files end with a .rtf extension.

## HTML Files (.htm)

StatPac creates standard HTML pages for a web survey. Multiple pages are created for multiple page surveys. These are initially created on your local computer. They are ordinary HTML pages and may be edited and enhanced with any HTML editor (e.g., Front Page). After you are satisfied with the appearance of the survey web pages, they are uploaded to your web site. The survey pages end with a .htm extension. You can also use your own editor to create HTML stylized email invitations.

## Perl Script (.pl)

When a respondent completes a page of a web survey, their answers are stored in a response file on your server. You will have one response file for each survey you host. The Perl script is a program installed on your server that controls the storage of the data. A Perl script is often called a CGI program. There are two Perl scripts provided with the software: *statpac11.pl* and *password.pl*. When you create an Internet survey, *statpac.pl* will be renamed to *yoursurveyname.pl* so that respondents see a meaningful URL in their browser address bar. The *password.pl* script is only used if you create a survey with password protection.

## Password Files (.text)

When creating a Web survey where each respondent has their own password, a tab delimited text file of valid passwords must be provided. While your password file can have any extension, StatPac will create a duplicate file with a .text extension. This file will be uploaded to the server to provide password access to the survey.

## Exported Data Files (.txt and .csv and .mdb)

After the data has been captured, you may want to export it to another format so a different program can use it. StatPac lets you export to tab delimited (.txt), comma delimited (.csv), and Access data bases (.mdb). These common formats can then be imported into most other software.

## Email Body Files (.txt or .htm)

When you send e-mail invitations to potential respondents, the body of the email is extracted from an ASCII text file during the mailing. The body of the e-mail can be created using any text editor (e.g., Microsoft Word, Notepad, etc.) and should be saved as a DOS text file. Usually, the extension will be .txt, but any extension is acceptable. Alternatively, the e-mail body may be a HTML file with a .htm extension.

## Sample File Naming Scheme for a Survey

In most situations, most of the files for a given survey will use the same file names and only the extensions will be different. Try to use file names that convey meaning to you. For example, if we wanted to call our survey “opinion”, we would have files called:

<code>opinion.cod</code>	(codebook)
<code>opinion.frm</code>	(form for manual data entry/editing)
<code>opinion.asc</code>	(ASCII text file of responses to a Web survey)
<code>opinion.dat</code>	(data file)
<code>opinion.pro</code>	(procedure file for performing analyses)
<code>opinion.htm</code>	(loader page of Web page)
<code>opinion_1.htm</code>	(first user viewable page of a Web survey)
<code>opinion_2.htm</code>	(another page of a Web survey or thank-you page)
<code>opinion.pl</code>	(Perl script created by StatPac to control page submissions)

We might also have (or create) several other files, but their file names do not have to be the same. Examples might be:

<code>something.rtf</code>	(a MS Word document used to facilitate the codebook design)
<code>opinion-mail.lst</code>	(an ASCII text file of e-mail addresses for the first mailing)
<code>opinion-mail-reminder.lst</code>	(an ASCII text file of e-mail addresses of nonresponders)

**opinion-body.txt** (an ASCII text of that contains the e-mail body for the first mailing) or **opinion-body.htm** if you were sending an HTML email instead of plain text

**opinion-body-reminder.txt** (an ASCII text of that contains the e-mail body for the second mailing to the nonresponders) or **opinion-body-reminder.htm** if you were sending an HTML emails instead of plain text

**opinion.txt** (an exported tab delimited file of the data)

**opinion.text** (a password file created control Web survey access)

---

## Customizing the Package

StatPac stores all its default values in a file called StatPac.ini. To a large degree, these settings control the operation of StatPac. This file can be found in the StatPac programs folder, which is usually C:\StatPac. However, if you installed StatPac in a different folder, it will be there instead.

Most of the settings in the StatPac.ini file are adjusted automatically when you run the program. However, there are some settings that you may wish to change manually. These are noted in this manual when applicable.

Use care when manually editing the StatPac.ini file. Some information (e.g., passwords) is encrypted in the file and will appear “funny” on the screen. Do not change these. Other settings may be crucial to the proper operation of the package. We suggest only changing the settings that are mentioned elsewhere in this manual.

The StatPac.ini file may be edited by selecting File, Open, System Defaults File. This will open the StatPac.ini file in the workspace area. Make the desired change(s) and click on the save icon (diskette), or select File, Save Workspace.

The StatPac.ini file is an ASCII text file, so it may alternatively be edited with any text editor (Notepad, WordPad, MS Word, etc.). To make a change in the StatPac.ini file using your own editor:

- 1) Close the StatPac for Windows program.
- 2) Load the StatPac.ini file using any text editor.
- 3) Make the change and save the file.

Most users will never need to directly edit the StatPac.ini file because nearly all the parameters can be changed within the program itself by setting analysis options or changing values in various settings windows. Various references in this manual may refer to other settings you might wish to alter.





# Basic Research Concepts

---

## Problem Recognition and Definition

We understand the world by asking questions and searching for answers. Our construction of reality depends on the nature of our inquiry.

All research begins with a question. Intellectual curiosity is often the foundation for scholarly inquiry. Some questions are not testable. The classic philosophical example is to ask, "How many angels can dance on the head of a pin?" While the question might elicit profound and thoughtful revelations, it clearly cannot be tested with an empirical experiment. Prior to Descartes, this is precisely the kind of question that would engage the minds of learned men. Their answers came from within. The scientific method precludes asking questions that cannot be empirically tested. If the angels cannot be observed or detected, the question is considered inappropriate for scholarly research.

Defining the goals and objectives of a research project is one of the most important steps in the research process. Do not underestimate the importance of this step. Clearly stated goals keep a research project focused. The process of goal definition usually begins by writing down the broad and general goals of the study. As the process continues, the goals become more clearly defined and the research issues are narrowed.

Exploratory research (e.g., literature reviews, talking to people, and focus groups) goes hand-in-hand with the goal clarification process. The literature review is especially important because it obviates the need to reinvent the wheel for every new research question. More importantly, it gives researchers the opportunity to build on each other's work.

The research question itself can be stated as a hypothesis. A hypothesis is simply the investigator's belief about a problem. Typically, a researcher formulates an opinion during the literature review process. The process of reviewing other scholar's work often clarifies the theoretical issues associated with the research question. It also can help to elucidate the significance of the issues to the research community.

The hypothesis is converted into a null hypothesis in order to make it testable because the only way to test a hypothesis is to eliminate alternatives of the hypothesis. Statistical techniques will enable us to reject or fail to reject a null hypothesis, but they do not provide us with a way to accept a hypothesis. Therefore, all hypothesis testing is indirect.

---

## Creating the Research Design

Defining a research problem provides a format for further investigation. A well-defined problem points to a method of investigation. There is no one best method of research for all situations. Rather, there are a wide variety of techniques for the researcher to choose from. Often, the selection of a technique involves a series of trade-offs. For example, there is often a trade-off between cost and the quality of information obtained. Time constraints sometimes force a trade-off with the overall research design. Budget and time constraints must always be considered as part of the design process.

---

## Methods of Research

There are three basic methods of research: 1) survey, 2) observation, and 3) experiment. Each method has its advantages and disadvantages.

The *survey* is the most common method of gathering information in the social sciences. It can be a face-to-face interview, telephone, mail, e-mail, or web survey. A personal interview is one of the best methods obtaining personal, detailed, or in-depth information. It usually involves a lengthy questionnaire that the interviewer fills out while asking questions. It allows for extensive probing by the interviewer and gives respondents the ability to elaborate their answers. Telephone interviews are similar to face-to-face interviews. They are more efficient in terms of time and cost, however, they are limited in the amount of in-depth probing that can be accomplished, and the amount of time that can be allocated to the interview. A mail survey is more cost effective than interview methods. The researcher can obtain opinions, but trying to meaningfully probe opinions is very difficult. Email and web surveys are the most cost effective and fastest methods.

*Observation* research monitors respondents' actions without directly interacting with them. It has been used for many years by A.C. Nielsen to monitor television viewing habits. Psychologists often use one-way mirrors to study behavior. Anthropologists and social scientists often study societal and group behaviors by simply observing them. The fastest growing form of observation research has been made possible by the bar code scanners at cash registers, where purchasing habits of consumers can now be automatically monitored and summarized.

In an *experiment*, the investigator changes one or more variables over the course of the research. When all other variables are held constant (except the one being manipulated), changes in the dependent variable can be explained by the change in the independent variable. It is usually very difficult to control all the variables in the environment. Therefore, experiments are generally restricted to laboratory models where the investigator has more control over all the variables.

---

## Sampling

It is incumbent on the researcher to clearly define the target population. There are no strict rules to follow, and the researcher must rely on logic and judgment. The population is defined in keeping with the objectives of the study.

Sometimes, the entire population will be sufficiently small, and the researcher can include the entire population in the study. This type of research is called a *census* study because data is gathered on every member of the population.

Usually, the population is too large for the researcher to attempt to survey all of its members. A small, but carefully chosen *sample* can be used to represent the population. The sample reflects the characteristics of the population from which it is drawn.

Sampling methods are classified as either *probability* or *nonprobability*. In probability samples, each member of the population has a *known non-zero* probability of being selected. Probability methods include random sampling, systematic sampling, and stratified sampling. In nonprobability sampling, members are selected from the population in some nonrandom manner. These include convenience sampling, judgment sampling, quota sampling, and snowball sampling. The advantage of probability sampling is that *sampling error* can be calculated. Sampling error is the degree to which a sample might differ from the population. When inferring to the population, results are reported plus or minus the sampling error. In nonprobability sampling, the degree to which the sample differs from the population remains unknown.

*Random sampling* is the purest form of probability sampling. Each member of the population has an equal and known chance of being selected. When there are very large populations, it is often difficult or impossible to identify every member of the population, so the pool of available subjects becomes biased.

*Systematic sampling* is often used instead of random sampling. It is also called an *Nth name selection* technique. After the required sample size has been calculated, every Nth record is selected from a list of population members. As long as the list does not contain any hidden order, this sampling method is as good as the random sampling method. Its only advantage over the random sampling technique is simplicity. Systematic sampling is frequently used to select a specified number of records from a computer file.

*Stratified sampling* is commonly used probability method that is superior to random sampling because it reduces sampling error. A *stratum* is a subset of the population that shares at least one common characteristic. The researcher first identifies the relevant strata and their actual representation in the population. Random sampling is then used to select subjects from each stratum until the number of subjects in that stratum is proportional to its frequency in the population. Stratified sampling is often used when one or more of the strata in the population have a low incidence relative to the other strata.

*Convenience sampling* is used in exploratory research where the researcher is interested in getting an inexpensive approximation of the truth. As the name implies, the sample is selected because they are convenient. This nonprobability method is often used during preliminary research efforts to get a gross estimate of the results, without incurring the cost or time required to select a random sample.

*Judgment sampling* is a common nonprobability method. The researcher selects the sample based on judgment. This is usually an extension of convenience sampling. For example, a researcher may decide to draw the entire sample from one "representative" city, even though the population includes all cities. When using this method, the researcher must be confident that the chosen sample is truly representative of the entire population.

*Quota sampling* is the nonprobability equivalent of stratified sampling. Like stratified sampling, the researcher first identifies the strata and their proportions as they are represented in the population. Then convenience or judgment sampling is used to select the required number of subjects from each stratum. This differs from stratified sampling, where the strata are filled by random sampling.

*Snowball sampling* is a special nonprobability method used when the desired sample characteristic is rare. It may be extremely difficult or cost prohibitive to locate

respondents in these situations. Snowball sampling relies on referrals from initial subjects to generate additional subjects. While this technique can dramatically lower search costs, it comes at the expense of introducing bias because the technique itself reduces the likelihood that the sample will represent a good cross section from the population.

---

## Data Collection

There are very few hard and fast rules to define the task of data collection. Each research project uses a data collection technique appropriate to the particular research methodology. The two primary goals for both quantitative and qualitative studies are to maximize response and maximize accuracy.

When using an outside data collection service, researchers often *validate* the data collection process by contacting a percentage of the respondents to verify that they were actually interviewed. Data *editing* and *cleaning* involves the process of checking for inadvertent errors in the data. This usually entails using a computer to check for out-of-bounds data.

*Quantitative* studies employ deductive logic, where the researcher starts with a hypothesis, and then collects data to confirm or refute the hypothesis. *Qualitative* studies use inductive logic, where the researcher first designs a study and then develops a hypothesis or theory to explain the results of the analysis.

Quantitative analysis is generally fast and inexpensive. A wide assortment of statistical techniques is available to the researcher. Computer software is readily available to provide both basic and advanced multivariate analysis. The researcher simply follows the preplanned analysis process, without making subjective decisions about the data. For this reason, quantitative studies are usually easier to execute than qualitative studies.

Qualitative studies nearly always involve in-person interviews, and are therefore very labor intensive and costly. They rely heavily on a researcher's ability to exclude personal biases. The interpretation of qualitative data is often highly subjective, and different researchers can reach different conclusions from the same data. However, the goal of qualitative research is to develop a hypothesis--not to test one.

Qualitative studies have merit in that they provide broad, general theories that can be examined in future research.

---

## Reporting the Results

The most important consideration in preparing any research report is the nature of the audience. The purpose is to communicate information, and therefore, the report should be prepared specifically for the readers of the report. Sometimes the format for the report will be defined for the researcher (e.g., a thesis or dissertation), while other times, the researcher will have complete latitude regarding the structure of the report. At a minimum, the report should contain an abstract, problem statement, methods section, results section, discussion of the results, and a list of references.

---

## Validity

*Validity* refers to the accuracy or truthfulness of a measurement. Are we measuring what we think we are? This is a simple concept, but in reality, it is extremely difficult to determine if a measure is valid.

Face validity is based solely on the judgment of the researcher. Each question is scrutinized and modified until the researcher is satisfied that it is an accurate measure of the desired construct. The determination of face validity is based on the subjective opinion of the researcher.

*Content validity* is similar to face validity in that it relies on the judgment of the researcher. However, where face validity only evaluates the individual items on an instrument, content validity goes further in that it attempts to determine if an instrument provides adequate coverage of a topic. Expert opinions, literature searches, and open-ended pretest questions help to establish content validity.

*Criterion-related validity* can be either predictive or concurrent. When a dependent/independent relationship has been established between two or more variables, criterion-related validity can be assessed. A mathematical model is developed to be able to predict the dependent variable from the independent variable(s). *Predictive validity* refers to the ability of an independent variable (or group of variables) to predict a future value of the dependent variable. *Concurrent validity* is concerned with the relationship between two or more variables at the same point in time.

*Construct validity* refers to the theoretical foundations underlying a particular scale or measurement. It explores the underlying theories or constructs that explain a phenomenon. This is also quite subjective and depends heavily on the understanding, opinions, and biases of the researcher.

---

## Reliability

*Reliability* is synonymous with repeatability. A measurement that yields consistent results over time is said to be reliable. When a measurement is prone to random error, it lacks reliability. The reliability of an instrument places an upper limit on its validity. A measurement that lacks reliability will necessarily be invalid. There are three basic methods to test reliability: test-retest, equivalent form, and internal consistency.

A *test-retest* measure of reliability can be obtained by administering the same instrument to the same group of people at two different points in time. The degree to which both administrations are in agreement is a measure of the reliability of the instrument. This technique for assessing reliability suffers two possible drawbacks. First, a person may have changed between the first and second measurement. Second, the initial administration of an instrument might in itself induce a person to answer differently on the second administration.

The second method of determining reliability is called the *equivalent-form* technique. The researcher creates two different instruments designed to measure identical constructs. The degree of correlation between the instruments is a measure of equivalent-form reliability. The difficulty in using this method is that it may be very difficult (and/or prohibitively expensive) to create a totally equivalent instrument.

The most popular methods of estimating reliability use measures of *internal consistency*. When an instrument includes a series of questions designed to examine the same construct, the questions can be arbitrarily split into two groups. The correlation between the two subsets of questions is called the *split-half* reliability. The problem is that this measure of reliability changes depending on how the questions are split. A better statistic, known as Cronbach's alpha, is based on the mean (absolute value) interitem correlation for all possible variable pairs. It provides a conservative estimate of reliability, and generally represents the lower bound to the reliability of a scale of items. For dichotomous nominal data, the KR-20 (Kuder-Richardson) is used instead of Cronbach's alpha.

---

## Systematic and Random Error

Most research is an attempt to understand and explain *variability*. When a measurement lacks variability, no statistical tests can be (or need be) performed. Variability refers to the dispersion of scores.

Ideally, when a researcher finds differences between respondents, they are due to true difference on the variable being measured. However, the combination of systematic and random errors can dilute the accuracy of a measurement. *Systematic error* is introduced through a constant bias in a measurement. It can usually be traced to a fault in the sampling procedure or in the design of a questionnaire. *Random error* does not occur in any consistent pattern, and it is not controllable by the researcher.

---

## Formulating Hypotheses from Research Questions

There are basically two kinds of research questions: testable and non-testable. Neither is better than the other, and both have a place in applied research.

Examples of non-testable questions are:

*How do managers feel about the reorganization?*

*What do residents feel are the most important problems facing the community?*

Respondents' answers to these questions could be summarized in descriptive tables and the results might be extremely valuable to administrators and planners. Business and social science researchers often ask non-testable research questions. The shortcoming with these types of questions is that they do not provide objective cut-off points for decision-makers.

In order to overcome this problem, researchers often seek to answer one or more testable research questions. Nearly all testable research questions begin with one of the following two phrases:

*Is there a significant difference between ...?*

*Is there a significant relationship between ...?*

For example:

*Is there a significant relationship between the age of managers and their attitudes towards the reorganization?*

*Is there a significant difference between white and minority residents with respect to what they feel are the most important problems facing the community?*

A research hypothesis is a testable statement of opinion. It is created from the research question by replacing the words "*Is there*" with the words "*There is*", and also replacing the question mark with a period. The hypotheses for the two sample research questions would be:

*There is a significant relationship between the age of managers and their attitudes towards the reorganization.*

*There is a significant difference between white and minority residents with respect to what they feel are the most important problems facing the community.*

It is not possible to test a hypothesis directly. Instead, you must turn the hypothesis into a null hypothesis. The null hypothesis is created from the hypothesis by adding

the words "no" or "not" to the statement. For example, the null hypotheses for the two examples would be:

*There is no significant relationship between the age of managers and their attitudes towards the reorganization.*

*There is no significant difference between white and minority residents with respect to what they feel are the most important problems facing the community.*

All statistical testing is done on the null hypothesis...never the hypothesis. The result of a statistical test will enable you to either 1) reject the null hypothesis, or 2) fail to reject the null hypothesis. Never use the words "accept the null hypothesis".

---

## Type I and Type II Errors

There are two types of hypothesis testing errors. The first one is called a *Type I error*. This is a very serious error where you wrongly reject the null hypothesis. Suppose that the null hypothesis is: Daily administrations of drug ABC will not help patients. Also suppose that drug ABC is really a very bad drug, and it causes permanent brain damage to people over 60. In your research, you ask for volunteers, and the entire sample is under 60 years of age. The sample seems to improve and you reject the null hypothesis. There could be very serious consequences if you were to market this drug (based on your sample). Type I errors are often caused by sampling problems.

A *Type II error* is less serious, where you wrongly fail to reject the null hypothesis. Suppose that drug ABC really isn't harmful and does actually help many patients, but several of your volunteers develop severe and persistent psychosomatic symptoms. You would probably not market the drug because of the potential for long-lasting side effects. Usually, the consequences of a Type II error will be less serious than a Type I error.

---

## Types of Data

One of the most important concepts in statistical testing is to understand the four basic types of data: nominal, ordinal, interval, and ratio. The kinds of statistical tests that can be performed depend upon the type of data you have. Different statistical tests are used for different types of data.

Nominal and ordinal data are *nonparametric* (non-continuous or categorical). Interval and ratio scales are called *parametric* (continuous). Some statistical tests are called parametric tests because they use parametric data. Others are called nonparametric tests because they use nonparametric data. All statistical tests are designed to be used with a specific kind of data, and may only be performed when you have that kind of data.

### ***Nominal data***

Nominal data is characterized by non-ordered response categories.

#### **Examples of nominal data**

What is your sex?

\_\_\_\_ Male    \_\_\_\_ Female

What program are you in?

\_\_\_\_ Administration/Management

☐ Health Services

☐ Education

☐ Human Services

Do you have health insurance?

☐ Yes   ☐ No   ☐ Don't know

What school did you attend?

☐ Park Elementary

☐ West Side

☐ Other

What should be done with the program?

☐ Close it down

☐ Seek government funding

☐ Hold a private fund raiser

☐ Other

What state do you live in? \_\_\_\_\_

Note: This question is called an *open-ended* question because it calls for a verbatim response. Even though the categories (i.e., the states) are not listed, the question is still considered nominal because the data can be categorized after it is collected.

Which of the following meats have you eaten in the last week? (Check all that apply)

☐ Hamburger   ☐ Pot roast   ☐ Liver

☐ Hotdogs   ☐ Bacon   ☐ Steak

☐ Pork chops   ☐ Sausage   ☐ Other

Note: This question is called a *multiple response* item because respondents can check more than one category. Multiple response simply means that a respondent can make more than one response to the same question. The data is still nominal because the responses are non-ordered categories.

What are the two most important issues facing our country today?

\_\_\_\_\_ and \_\_\_\_\_

Note: This question is an open-ended multiple response item because it calls for two verbatim responses. It is still considered nominal data because the issues could be categorized after the data is collected.

## ***Ordinal data***

Ordinal data is characterized by ordered response categories.

### **Examples of ordinal data**

What is your highest level of education?

☐ Grade school

☐ Some high school

☐ High school graduate

☐ Some college



\_\_\_ College graduate

\_\_\_ Advanced degree

How many beers have you drunk in the last week?

\_\_\_ None    \_\_\_ One to five    \_\_\_ Six to ten    \_\_\_ Over ten

How would you rate your progress?

\_\_\_ Excellent

\_\_\_ Good

\_\_\_ Fair

\_\_\_ Poor

What has the trend been in your business over the past year?

\_\_\_ Decreasing    \_\_\_ Stable    \_\_\_ Increasing

Please rate the quality of this lecture?

\_\_\_ Low    \_\_\_ Medium    \_\_\_ High

Use a circle to indicate your level of agreement or disagreement with the following statement: Abortion should be a decision between a woman and her doctor.

Strongly				Strongly
Agree	Agree	Neutral	Disagree	Disagree
1	2	3	4	5

What is your annual family income?

\_\_\_ Under \$12,000

\_\_\_ \$12,000 to \$23,999

\_\_\_ \$24,000 to \$49,999

\_\_\_ \$50,000 to \$74,999

\_\_\_ \$75,000 or more

### ***Interval and ratio data***

Interval and ratio data are such that each numeric *interval* represents one unit of measurement. *Ratio* scales also have the property of an absolute "zero-point". *Interval* and *ratio-scaled* questions are preferable in research design because they offer the most versatility in the kinds of analyses that may be performed.

#### **Examples of interval and ratio data**

What is your age? \_\_\_\_\_

How many children do you have? \_\_\_\_\_

What was your SAT score? \_\_\_\_\_

How many years of school have you completed? \_\_\_\_\_

What percent of your work time do you spend .... ? \_\_\_\_\_

How many collective bargaining sessions have you been involved in? \_\_\_\_\_

What is the average class size in your school? \_\_\_\_\_

What was your family income last year? \_\_\_\_\_

How many units have you completed? (Circle) 0 1 2 3

What was your GPA as an undergraduate student? \_\_\_\_\_

How many times have you been arrested? \_\_\_\_\_

---

## Significance

What does significance really mean?

Many researchers get very excited when they have discovered a "significant" finding, without really understanding what it means. When a statistic is significant, it simply means that you are very sure that the statistic is reliable. It doesn't mean the finding is important.

For example, suppose we give 1,000 people an IQ test, and we ask if there is a significant difference between male and female scores. The mean score for males is 98 and the mean score for females is 100. We use an independent groups t-test and find that the difference is significant at the .001 level. The big question is, "So what?" The difference between 98 and 100 on an IQ test is a very small difference...so small, in fact, that it's not even important.

Then why did the t-statistic come out significant? Because there was a large sample size. When you have a large sample size, very small differences will be detected as significant. This means that you are very sure that the difference is real (i.e., it didn't happen by fluke). It doesn't mean that the difference is large or important. If we had only given the IQ test to 25 people instead of 1,000, the two-point difference between males and females would not have been significant.

Significance is a statistical term that tells how sure you are that a difference or relationship exists. To say that a significant difference or relationship exists only tells half the story. We might be very sure that a relationship exists, but is it a strong, moderate, or weak relationship? After finding a significant relationship, it is important to evaluate its strength. Significant relationships can be strong or weak. Significant differences can be large or small. It just depends on your sample size.

Many researchers use the word "significant" to describe a finding that may have decision-making utility to a client. From a statistician's viewpoint, this is an incorrect use of the word. However, the word "significant" has virtually universal meaning to the public. Thus, many researchers use the word "significant" to describe a difference or relationship that may be strategically important to a client (regardless of any statistical tests). In these situations, the word "significant" is used to advise a client to take note of a particular difference or relationship because it may be relevant to the company's strategic plan. The word "significant" is not the exclusive domain of statisticians and either use is correct in the business world. Thus, for the statistician, it may be wise to adopt a policy of always referring to "statistical significance" rather than simply "significance" when communicating with the public.

---

## One-Tailed and Two-Tailed Tests

One important concept in significance testing is whether to use a one-tailed or two-tailed test of significance. The answer is that it depends on your hypothesis. When your research hypothesis states (or implies) the direction of the difference or relationship, then you use a one-tailed probability. For example, a one-tailed test would be used to test these null hypotheses: Females will not score significantly higher than males on an IQ test. Blue collar workers will not have significantly lower education than white collar workers. Superman is not significantly stronger

than the average person. In each case, the null hypothesis (indirectly) predicts the direction of the expected difference. A two-tailed test would be used to test these null hypotheses: There will be no significant difference in IQ scores between males and females. There will be no significant difference between blue collar and white collar workers. There is no significant difference in strength between Superman and the average person. A one-tailed probability is exactly half the value of a two-tailed probability.

There is a raging controversy (for about the last hundred years) on whether or not it is ever appropriate to use a one-tailed test. The rationale is that if you already know the direction of the difference, why bother doing any statistical tests. The safest bet is to always state your hypotheses so that two-tailed tests are appropriate.

---

## Procedure for Significance Testing

Whenever we perform a significance test, it involves comparing a test value that we have calculated to some critical value for the statistic. It doesn't matter what type of statistic we are calculating (e.g., a t-statistic, a chi-square statistic, an F-statistic, etc.), the procedure to test for significance is the same.

1. Decide on the *critical alpha level* ( $\alpha$ ) you will use (i.e., the error rate you are willing to accept).
2. Conduct the research.
3. Calculate the statistic.
4. Compare the statistic to a *critical value* obtained from a table or compares the probability of the statistic to the *critical alpha level*.

If your statistic is higher than the *critical value* from the table or the probability of the statistic is less than the critical alpha level:

Your finding is significant.

You reject the null hypothesis.

The probability is small that the difference or relationship happened by chance, and  $p$  is less than the critical alpha level ( $p < \alpha$ ).

If your statistic is lower than the *critical value* from the table or the probability of the statistic is higher than the critical alpha level:

Your finding is not significant.

You fail to reject the null hypothesis.

The probability is high that the difference or relationship happened by chance, and  $p$  is greater than the critical alpha level ( $p > \alpha$ ).

Modern computer software can calculate exact probabilities for most test statistics. When StatPac (or other software) gives you an exact probability, simply compare it to your critical alpha level. If the exact probability is less than the critical alpha level, your finding is significant, and if the exact probability is greater than your critical alpha level, your finding is not significant. Using a table is not necessary when you have the exact probability for a statistic.

---

## Bonferroni's Theorem

Bonferroni's theorem states that as one performs an increasing number of statistical tests, the likelihood of getting an erroneous significant finding (Type I error) also increases. Thus, as we perform more and more statistical tests, it becomes increasingly likely that we will falsely reject a null hypothesis (very bad).

For example, suppose our critical alpha level is .05. If we performed one statistical test, our chance of making a false statement is .05. If we were to perform 100 statistical tests, and we made a statement about the result of each test, we would expect five of them to be wrong (just by fluke). This is a rather undesirable situation for social scientist.

Bonferroni's theorem states that we need to adjust the critical alpha level in order to compensate for the fact that we're doing more than one test. To make the adjustment, take the desired critical alpha level (e.g., .05) and divide by the number of tests being performed, and use the result as the critical alpha level. For example, suppose we had a test with eight scales, and we plan to compare males and females on each of the scales using an independent groups t-test. We would use .00625 (.05/8) as the critical alpha level for all eight tests.

Bonferroni's theorem should be applied whenever you are conducting two or more tests that are of the same "type" and the same "family". The same "type" means the same kind of statistical test. For example, if you were going to do one t-test, one ANOVA, and one regression, you would not make the adjustment because the tests are all different. The same "family" is a more elusive concept, and there are no hard and fast rules. "Family" refers to a series of statistical tests all designed to test the same (or very closely related) theoretical constructs. The bottom line is that it's up to the individual researcher to decide what constitutes a "family".

Some things are more obvious than others, for example, if you were doing t-tests comparing males and females on a series of questionnaire items that are all part of the same scale, you would probably apply the adjustment, by dividing your critical alpha level by the number of items in the scale (i.e., the number of t-tests you performed on that scale). The probabilities of the tests would be called the *family error rates*. However, suppose you have a series of independent questions, each focusing on a different construct and you want to compare males and females on how they answered each question. Here is where the whole idea of Bonferroni's adjustment becomes philosophical. If you claim that each t-test that you perform is a test of a unique "mini"-hypothesis, then you would not use the adjustment, because you have defined each question as a different "family". In this case, the probability would be called a *statement error rate*. Another researcher might call the entire questionnaire a "family", and she would divide the critical alpha by the total number of items on the questionnaire.

Why stop there? From a statistician's perspective, the situation becomes even more complex. Since they are personally in the "statistics business", what should they call a "family"? When a statistician does a t-test for a client, maybe she should be dividing the critical alpha by the total number of t-tests that she has done in her life, since that is a way of looking at her "family". Of course, this would result in a different adjustment for each statistician--an interesting dilemma.

In the real world, most researchers do not use Bonferroni's adjustment because they would rarely be able to reject a null hypothesis. They would be so concerned about the possibility of making a false statement, that they would overlook many differences and relationships that actually exist. The "prime directive" for social science research is to discover relationships. One could argue that it is better to risk making a few wrong statements, than to overlook relationships or differences that are

clear or prominent, but don't meet critical alpha significance level after applying Bonferroni's adjustment.

---

## Central Tendency

The best known measures of central tendency are the mean and median. The mean average is found by adding the values for all the cases and dividing by the number of cases. For example, to find the mean age of all your friends, add all their ages together and divide by the number of friends. The mean average can present a distorted picture of central tendency if the sample is skewed in any way.

For example, let's say five people take a test. Their scores are 10, 12, 14, 18, and 94. (The last person is a genius.) The mean would be the sums of the scores  $10+12+14+18+94$  divided by 5. In this example, a mean of 29.6 is not a good measure of how well people did on the test in general. When analyzing data, be careful of using only the mean average when the sample has a few very high or very low scores. These scores tend to skew the shape of the distribution and will distort the mean.

When you have sampled from the population, the mean of the sample is also your best estimate of the mean of the population. The actual mean of the population is unknown, but the mean of the sample is as good an estimate as we can get.

The median provides a measure of central tendency such that half the sample will be above it and half the sample will be below it. For skewed distributions this is a better measure of central tendency. In the previous example, 14 would be the median for the sample of five people. If there is no middle value (i.e., there are an even number of data points), the median is the value midway between the two middle values.

The distribution of many variables follows that of a bell-shaped curve. This is called a "normal distribution". One must assume that data is approximately normally distributed for many statistical analyses to be valid. When a distribution is normal, the mean and median will be equal to each other. If they are not equal, the distribution is distorted in some way.

---

## Variability

Variability is synonymous with diversity. The more diversity there is in a set of data, the greater the variability. One simple measure of diversity is the range (maximum value minus the minimum value). The range is generally not a good measure of variability because it can be severely affected by a single very low or high value in the data. A better method of describing the amount of variability is to talk about the dispersion of scores away from the mean.

The variance and standard deviation are useful statistics that measure the dispersion of scores around the mean. The standard deviation is simply the square root of the variance. Both statistics measure the amount of diversity in the data. The higher the statistics, the greater the diversity. On the average, 68 percent of all the scores in a sample will be within plus or minus one standard deviation of the mean and 95 percent of all scores will be within two standard deviations of the mean.

There are two formulas for the variance and standard deviation of a sample. One set of formulas calculates the exact variance and standard deviation of the sample. The statistics are called *biased*, because they are biased to the sample. They are the exact variance and standard deviation of the sample, but they tend to underestimate the variance and standard deviation of the population.

Generally, we are more concerned with describing the population rather than the sample. Our intent is to use the sample to describe the population. The *unbiased estimates* should be used when sampling from the population and inferring back to the population. They provide the best estimate of the variance and standard deviation of the population.

---

## Standard Error of the Mean

The standard error of the mean is used to estimate the range within which we would expect the mean to fall in repeated samples taken from the population (i.e., confidence intervals). The standard error of the mean is an estimate of the standard deviation of those repeated samples.

The formula for the standard error of the mean provides an accurate estimate when the sample is very small compared to the size of the population. In marketing research, this is usually the case since the populations are quite large. However, when the sample size represents a substantial portion of the population, the formula becomes inaccurate and must be corrected. The finite population correction factor is used to correct the estimate of the standard error when the sample is more than ten percent of the population.

---

## Inferences with Small Sample Sizes

When the sample size is small (less than 30), the  $z$  value for the area under the normal curve is not accurate. Instead of a  $z$  value, we can use a  $t$  value to derive the area under the curve. In fact, many researchers always use the  $t$  value instead of the  $z$  value. The reason is that the  $t$  values are more accurate for small sample sizes, and they are nearly identical to the  $z$  values for large sample sizes. Unlike the  $z$  value, the values for  $t$  depend upon the number of cases in the sample. Depending on the sample size, the  $t$  value will change.

---

## Degrees of Freedom

Degrees of freedom literally refers to the number of data values that are free to vary. For example, suppose I tell you that the mean of a sample is 10, and there are a total of three values in the sample. It turns out that if I tell you any two of the values, you will always be able to figure out the third value. If two of the values are 8 and 12, you can calculate that the third value is 10 using simple algebra.

$$(x + 8 + 12) / 3 = 10 \quad x = 10$$

In other words, if you know the mean, and all but one value, you can figure out the missing value. All the values except one are free to vary. One value is set once the others are known. Thus, degrees of freedom is equal to  $n-1$ .

# Codebook Design

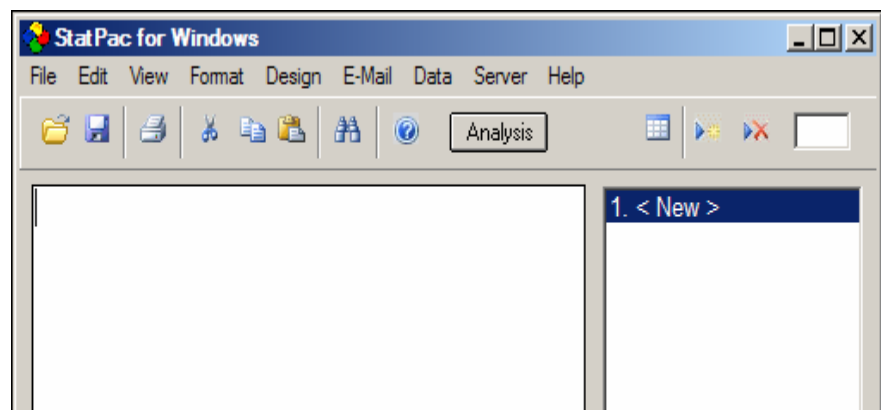
---

## Components of a Study Design

All surveys begin by creating a *codebook*. The codebook contains the format and labels of each variable. If your survey contains 20 items, the codebook will also contain 20 items.

If the survey will be administered by paper and pencil or CATI there will also be a data entry *form*. The form refers to the screens that the data entry person will see while entering and editing data. The codebook and form usually have the same file names. Only the file extensions are different.

If the survey will be administered over the Internet or as an e-mail survey, a data entry form is not necessary.



There are many ways to design the codebook and form. The best way depends upon whether or not you already have typed the survey with a word-processor. When you run the program, the main screen will be displayed. The left side of the screen is for the Workspace window and the right side of the screen is used to show the list of variables in the codebook.

---

## Elements of a Variable

The codebook defines the variables in the study. Each item on the survey is a variable. Thus, the number of variables in the codebook is the same as the number of items. (Note that several variables are required to define a multiple response item, so the number of variables in the codebook might actually exceed the number of items on the survey.)

There are five components to a variable. These are:

Variable Format

Variable Name

Variable Label

Value Labels (including valid codes and skip codes)

Data Entry Control Parameters

The Variable Format is mandatory because it defines where and how the data is stored in the data file. All other components are optional.

### Variable Format

The Variable Format defines the *structure* of the variable. This is the only information that is mandatory when defining a new variable. Once you define the structure of a variable, it will exist in the codebook.

The syntax for the Variable Format is:

<Var. Type> <No. of Cols> . <Decimal>

The following are examples of variable formats. The word “columns” refers to the number of characters that will be available to hold the respondent’s answer/:

N5      a numeric variable using 5 columns

N5.2    a numeric variable using 5 columns;  
         the format of the variable will be ##.##

N2      a numeric variable using 2 columns

A1      an alpha variable using 1 column

A250    an alpha variable using 250 columns

### Variable Type

StatPac has two types of variables: numeric and alpha. Different analyses can be performed depending on the variable type. StatPac requires that a variable type be specified as either N or A.

Numeric variables may contain numbers, a decimal point, a plus or minus sign and a D or E if scientific notation is used. Alpha variables may contain any character (letters, numbers and special characters). StatPac automatically left justifies alpha variables and right justifies numeric variables.

An example of a numeric variable might be the following question on a survey:

*How many years of formal education have you completed?*

The response would always be a number that could be contained in two columns of the data file. The response would also always be numeric. A numeric-type specification is required for interval or ratio-type data.



Some questions have coded responses and could use the alpha-type format. An example of alpha-type question on a survey would be:

*Which product do you prefer?*

*A = Product A*

*B = Product B*

*C = Product C*

*N = No preference*

In this question, the responses are coded into categories. The categories are not arithmetically related. That is, a response of C does not mean twice as much product as response A. Nominal and ordinal-type data can use either an alpha or numeric format.

Another example of an alpha variable would be an open-ended response. The respondent could answer anything to the following question:

*What could we do to improve our product?*

Likert-scale questions and preference scales are often given a numeric format so that descriptive statistics can be calculated. This is a generally accepted procedure in marketing and social science research, the assumption being that the perceived intervals between the selections are equal.

### **Number of Columns**

The number of columns component of the format statement is the field width allocated for the variable. This is the number of characters needed to write the longest data value. There is not a maximum number of columns for an alpha variable, although the practical limit for data entry is 1,000 - 2,000 characters. For a numeric variable, the maximum number of columns is 22 characters.

The field width for numeric variables must be large enough to hold the number, a plus or minus sign, and a decimal point (if necessary). For example, a numeric one-to-ten scale would require two characters; racing times for a hundred meter sprint (with accuracy to the hundredth of a second) would require five characters (two for the seconds, one for the decimal point, and two for the hundredths of seconds). An alpha variable to hold an entire open-ended sentence might require 150 characters.

It is very important that you leave a sufficient number of columns for your data. After you begin entering data, changing the number of columns for a variable will become more complex (since this requires restructuring of the data already entered). User Tip: When in doubt, allow more columns rather than less.

### **Decimal Places**

The decimal format is the number of significant decimal places that the variable will contain. This component of the format statement is optional and may be omitted. If *<decimal>* is not specified, the data will be stored exactly as entered (with or without a decimal point). In the format statement itself, the number of decimal digits is preceded by a decimal point.

### **Variable Name**

The variable name is simply a name that may be used to reference the variable when designing analyses. While the variable name is optional, its use is highly recommended. As a general rule, the variable name is a short word or abbreviation. Its primary purpose is to help you keep track of variables while designing analyses.

There are several rules governing variable names. All of these are automatically checked by StatPac so it will not be possible to enter an invalid variable name.

1. A variable name must be unique from all other variable names and may not be the same as any analysis keyword. (The keywords are listed in another section of the manual.)
2. The first character of a variable name may not be a number or a space.
3. A variable name may not be the same as a V number. For example, you cannot name a variable "V12".
4. A variable name may not contain a comma or period. The variable name may include a space; however, for the purpose of clarity, we recommend using an underscore character instead of a space.
5. A variable name may not be D, E, RECORD, TIME, LO, HI, WITH, BY, THEN, TOTAL or MEAN. These words have special meaning to StatPac.

## Variable Label

The variable label is a written description of the variable. For surveys, the variable label is usually the question itself. There are no restrictions on the content or length of a variable label. It may contain any character on the keyboard.

When creating a series of multiple response variables, identical variable labels should be used for each of the multiple response items. This tells StatPac how to format the data entry form, and thereby improves data entry.

## Value Labels

Value labels are used in the reports to label the response categories. They may include any upper or lower case character except a semicolon. The format for a value label is:

`<Code>=<Label>`

The code on the left of the equals sign is what will be typed during data entry. The label on the right of the equals sign is the *definition* of the code and will be used to label the output.

In the following example, there are four value labels. These are entered as four separate lines.

*1=6 years or less*  
*2=7 to 9 years*  
*3=10 to 12 years*  
*4=Over 12 years*

There are no spaces between the code and the equals sign. There are also no spaces between the equals sign and the label. The code on the left of the equals sign may not be greater than the field width defined in the format statement. There is not a limit on the length of the value label (on the right of the equals symbol); however, short value labels (20 or fewer characters) generally produce more condensed and easier to read printouts.

For alpha variables, it is important to note that upper and lower case characters are different. When you enter the code on the left of the equals sign, the code should be the same case as you plan to enter the data. For example, if the data entry person will be entering a lower case m and f for male and female, the value labels would be:

*m=Male*

*f=Female*

The value labels also define what will be accepted as valid data during data entry. **Whenever a value label is specified, the code (on the left of the equals sign) will be interpreted as a valid code during data entry. If no value labels are specified, all data will be considered valid.**

Many variables do not need any value labels. **They are required only when a coded response will be entered.** Numeric interval and ratio data, as well as open ended alpha data, do not require any value labels.

The following questions would not need value labels:

*What is your age?*

What is your first name?

What score did you get on the test?

What is your favorite number?

What would best describe your feelings?

## Valid Codes

When there are no value labels (such as a test score variable), valid codes for data entry can still be specified by simply typing the valid codes or ranges. The format for entering valid codes is:

*<Code or Range>*

*<Code or Range>*

*<Code or Range>*

In this case, each valid code (or valid range) is entered on a different line.

Alternately, a slash (/) may be used to list a series of valid codes on the same line:

*<Code or Range>/<Code or Range>/<Code or Range>*

The following examples illustrate various ways to specify valid codes.

*1/3/5*    *accept codes 1, 3 & 5*

*1-3*    *accept codes 1 to 3*

*1-3/5*    *accept 1 to 3 & 5*

*15-99*    *accept values 15 to 99*

*A-D*    *accept codes A, B, C and D*

*A-D/X*    *accept codes A, B, C, D and X*

*#*    *accept anything*

Notice that the pound symbol (#) is used to specify "accept any number or letter during data entry". If the field is numeric, this means any number is an acceptable value. If the field is alpha, it means that any character is acceptable input.

When the valid codes, labels, and skips field is completely empty, any input will be accepted (i.e., it is the same as the # symbol). If the pounds symbol is specified, it should be the last line of the value labels for the variable.

---

## Skip Codes for Branching

*Skip codes* allow you to specify conditions for passing over certain variables during data entry depending on the values entered for other variables. This is commonly referred to as *branching*.

For instance, if variable 6 contains responses to the question "Have you ever read Music Magazine?" and variable 7 stored answers to the question "How much do you like Music Magazine?", you would want to skip to variable 8 for a person who responded "No" to variable 6.

A semicolon and "branch to" number may be used on a *<Code>=<Label>* line to control data entry branching. For the Music Magazine example above, the value labels would be:

```
Y=Yes
N=No ;8
```

Note that the semicolon and variable to branch to follow the value label. In this example, the space before the semicolon is for readability only. All of the following lines are equivalent:

```
N=No;8
N=No      ;8
N=No      ;      8
```

As another example, consider a questionnaire that includes a "dwelling" variable for which 1=Apartment, 2=Condominium, and 3=House. If three separate sections within the questionnaire corresponded to each type of dwelling, the value labels and *skip* codes for the dwelling variable could be:

```
1=Apartment      ;14
2=Condominium    ;23
3=House          ;29
```

The *skip* codes would direct data entry to variables 14, 23 or 29 depending on whether a 1, 2 or 3 was entered. Again note that the spacing is for readability only.

The pound symbol (#) may be used in a *skip* code to mean any value or code. That is, it is an absolute jump to a variable regardless of the data entered. For example, *#=;14* means to jump to variable 14 after entering the current field. This feature is useful when you want to end a branch and rejoin with a common variable, as in the dwelling example above.

Complex branching is also supported. This means that a branch can be based on the response to a previous variable. The following is an example of how to use complex branching. Assume it is the value labels for variable 10. If the data for variable 10 is entered as a 1 or 2, the complex skip will be evaluated. In this example, the skip pattern is the same regardless of whether a 1 or 2 is entered. If the previous response to variable 5 was 1 then the skip will go to variable 25, and if the previous response to variable 5 was 2 then the skip will go to variable 30. A response of 3 for this variable would skip to variable 35.

```
1=Yes ; #V5=1 ; 25 #V5=2 ; 30
2=No  ; #V5=1 ; 25 #V5=2 ; 30
3=No Answer ; 35
```

Note that a semicolon is used to begin the complex skip and before each "skip to" variable. Also note that a pound symbol is used to start each portion of the complex skip. All spacing is optional. Complex skip patterns are not automatically updated if

you insert or delete a variable from the codebook. Therefore, they are generally added after the structure of the codebook has been finalized.

If you specify a *skip* to a nonexistent variable number, it will be interpreted as an instruction to branch to the end of the questionnaire. For example, if you have a survey with fifty questions, a *skip* to variable ninety-nine would mean to immediately end the current questionnaire, and begin a new interview with the next respondent.

Be careful when defining *skip* codes, as it is quite possible to create an endless data entry loop.

---

## Data Entry Control Parameters

The Data Entry Control parameters determine how the data entry program will operate. They can be set independently for each variable, and are all of the yes/no variety.

### Missing OK

The decision whether or not to allow missing data for a particular variable depends upon the variable itself. For example, ID number may be something you want to make mandatory during data entry (no missing data will be allowed). Some variables however, should accept missing data. For example, in surveys, respondents may leave questions blank or simply prefer not to answer others; in agricultural research, some of the crop dies; in public health research, participants move, etc. When in doubt, missing data should be allowed. This only means that the data entry person will be able to skip over this variable if they need to.

For Internet surveys, we strongly recommend allowing missing data for all variables. The ease in which a person can leave an Internet survey makes it exceedingly important that they not become frustrated by the process. Requiring input when a respondent does not wish to answer an item will most assuredly result in the partially completed survey.

### Auto Advance

When the Auto Advance is set, the cursor will automatically move to the next field when the current field is filled with characters. This means that during the data entry process, if you type the same number of characters that were reserved for field width (in the format statement), you will not need to press <enter> to move to the next field. This will significantly speed up the data entry process since it eliminates a keystroke (i.e., <enter>) for each variable. This parameter will be ignored for Internet surveys.

### Caps Only

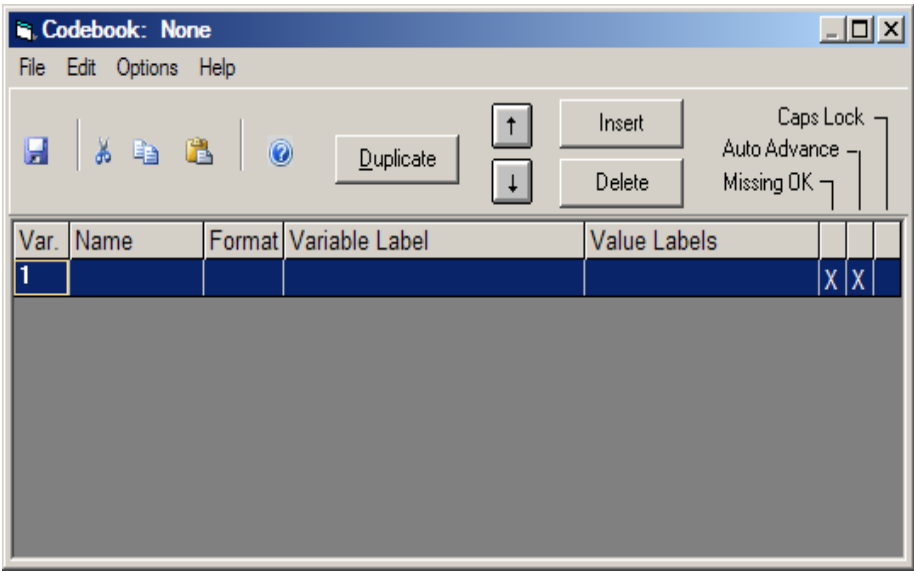
The Caps Only parameter determines whether the characters typed on the keyboard will be converted to upper case letters in the data file. This is especially useful if a field is coded alpha, and you do not want the data entry operator to be able to inadvertently enter lower case characters. It is identical to using the caps lock on your keyboard. This parameter will be ignored for Internet surveys.

# Codebook Tools

There are two main tools for entering and changing the information in the codebook: the Grid and the Variable Detail. Either tool may be used at any time. Generally, the Grid is used when you are beginning a new codebook, and the Variable Detail is used to make changes to individual variables. There is also an Analysis utility program “Quick Codebook Creation” to create a codebook from an extended format statement.

## The Grid

One method of designing a codebook is to use the Grid. Click on the Grid button and the Grid will be displayed.

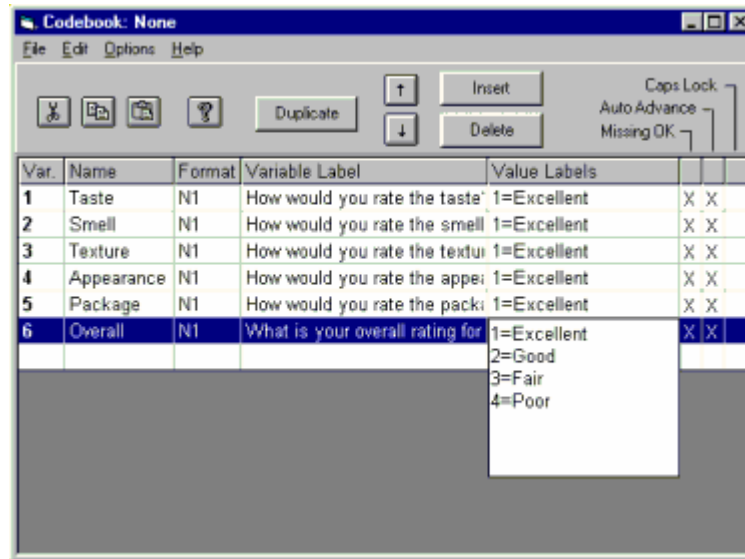


A row in the Grid represents a variable. If your study has 50 variables, there will be 50 rows in the Grid. When you start the Grid, only one row will be showing. More rows will appear as needed as you enter the codebook. When you enter a variable format for the current variable, a blank row for a new variable will appear.

To begin entering information into the Grid, click in the name field of the first row.

Use the Tab key and Shift Tab keys to move from one column to the next. You can also use the left or right mouse buttons to select a field.

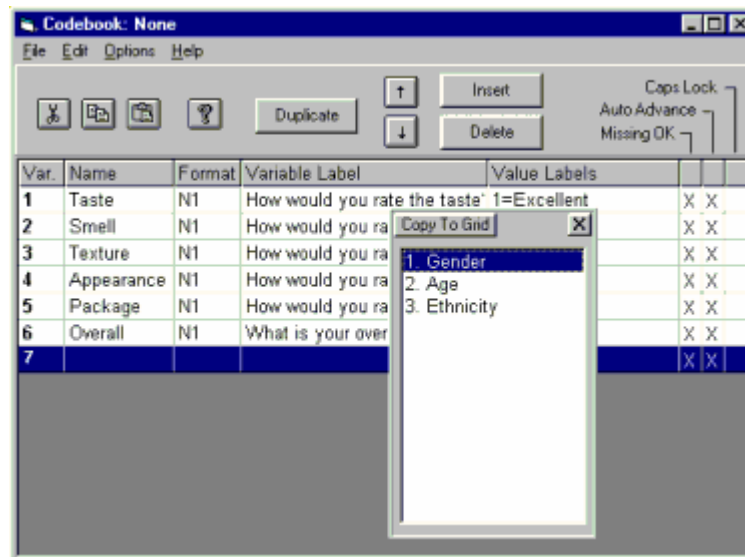
The Variable Label and Value Label fields will display a larger window when you enter those fields. If either of these windows is showing, you can minimize it by clicking on it with the right mouse button, or clicking on another field.



## Codebook Libraries

There are many features to make the codebook design easier. One of these is the ability to load variables from other codebooks. In other words, you can establish a "library" of commonly used questions. The library can be a codebook that you designed especially for this purpose, or it can be a codebook that you used for a previous study.

To load a variable or variables from a library, select File, Open Library.



After loading the library, you can choose one or more variables to copy to the new codebook. To select multiple variables, hold down the control key while you click on the individual variables in the library. After selecting the variables, click on the Copy To Grid Button in the top left corner of the library window.

## Duplicating Variables

Many times, consecutive variables in the codebook are similar. While working with the Grid, you can copy the information from the previous variable to the current variable. While entering a new variable, click on the Duplicate Button to repeat all the information from the previous variable. StatPac will automatically change the variable name since two variables cannot share the same name. StatPac will not duplicate any fields that are not blank in the current variable.

The Duplicate Button is especially useful when creating a series of variables that share the same value labels or a series of multiple response variables. For example, if you are entering a series of variables that all use the same value labels, you could enter the variable format, name, label, and then click the Duplicate button to repeat the value labels from the previous variable. When entering multiple response variables, you could use the Duplicate Button to repeat the entire variable.

The library feature can also be used to duplicate variables in the current codebook. Unlike the Duplicate Button (which duplicates only the previous variable), the library can be used to duplicate variable(s) that appear anywhere in the codebook. First save the codebook by clicking on the Save button or selecting File, Save Codebook. Then click on the row where you want the new variables to be inserted. Select File, Open Library and select the current codebook as the library. Finally select the variables you want to duplicate, and click the Copy To Grid Button.

## Insert & Delete Variables

Normally, while you are designing a study, variables are added one after another to the end of the existing variables. However, you can also insert a new variable in the middle of the codebook.

Click on the Grid row you want to be immediately below the new variable. Then click the Insert Button to open up a blank new row in the Grid.

To delete a variable, first click on the Grid row you want to delete. Then click the Delete Button.

## Move Variables

The order of the variables can be changed using the Up and Down Arrow Buttons. First, click on the variable you want to move. Then click the Up or Down Arrow Buttons to move the variable.

## Starting Columns

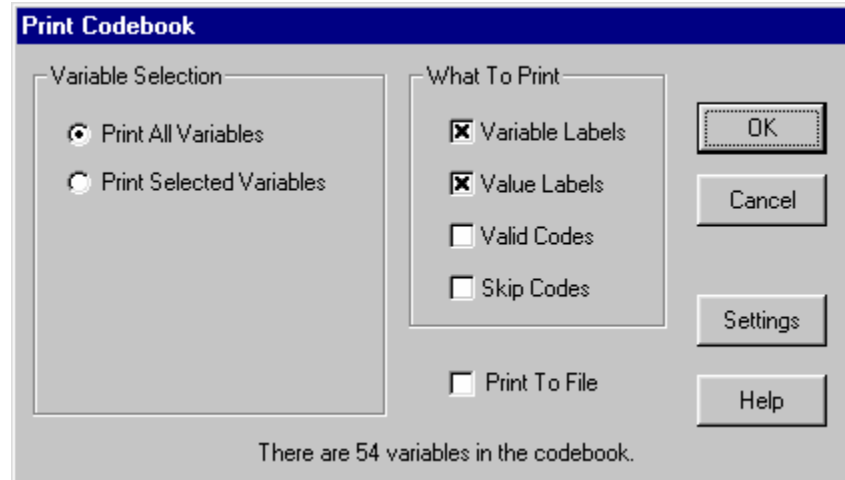
Starting columns refer to the beginning location of the variables in the data record. During data entry, each variable that is entered will be stored in the data record beginning at a certain location. The starting columns are these locations.

Starting columns are automatically determined, and you do not need to be concerned about them. That is, starting columns are assigned by the program while entering new variables into the Grid. They are assigned so the data record will store variables in consecutive (contiguous) columns. Thus, the starting columns are being automatically handled by the program and not displayed as part of the Grid. If necessary, they will be adjusted whenever the codebook is saved. They can be displayed by selecting Options, Show Start Columns.



## Print a Codebook

To print a codebook, select File, Print, Codebook. The Print Dialog window will give you the opportunity to choose various printing options. Printing a codebook is especially important if you give your data file to someone else, since the codebook will tell them exactly how the data is formatted.



The Variable Selection lets you select which variables from the codebook will be printed. The list of variables to print can use spaces or commas to separate variables, and dashes to indicate a range of variables.

A codebook printout will always include the variable numbers, names, and formats for the variables. The "What To Print" items let you select what additional information from the codebook will be printed.

Variable Labels - When this parameter is set, variable labels will be printed.

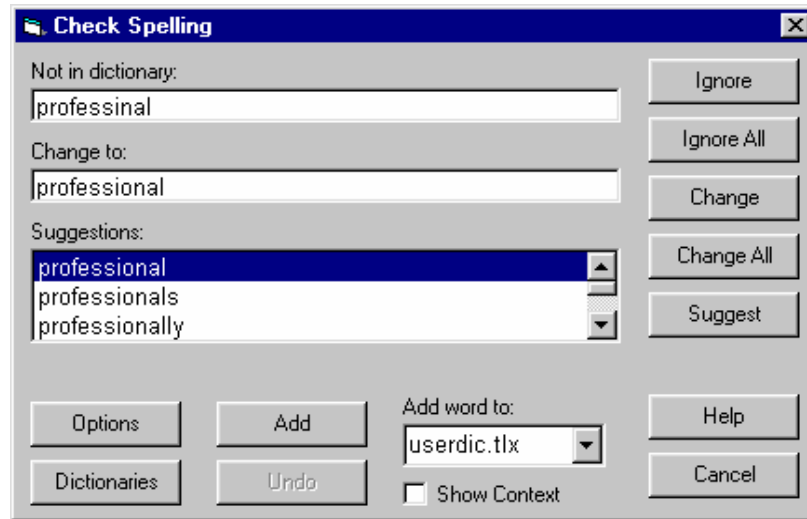
Value Labels - When this parameter is set, value labels will be printed.

Valid Codes - When this parameter is set, valid codes will be printed. This specifically refers to valid codes that are not part of a `<Code>=<Label>`.

Skip Codes- When this parameter is set, skip patterns will be printed as part of the value labels.

## Spell Check a Codebook

To check the spelling in a codebook, select Design, Spell Check. The spelling checker dialog box will be shown.



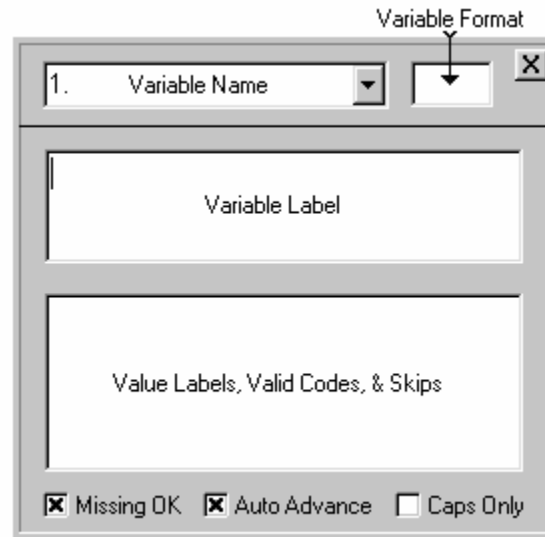
The default dictionary for the spelling check is American English. The software also includes spelling dictionaries for British English, French, Spanish, and German. To change the dictionary that StatPac uses, you must edit the StatPac.ini file. Find the line that says *DictionaryName = English*. Change the word "English" to "British", "French", "Spanish", or "German".

## Variable Detail Window

To show the Variable Detail window, select View, Variable Detail. If the Variable List window is showing, you can also double click on a variable to evoke the Variable Detail window. The variable window gives you the ability to add or modify nearly all the information in the codebook. While the layout is different, it gives you the same functionality as the Grid.

The Variable Detail window can be moved around the screen by pressing the mouse button on any gray area of the window and dragging the window to a new location.

When you change any of the fields in the Variable Detail window, the change is instantly reflected in the codebook. See Elements of a Variable for a complete description of each field.




---

## Codebook Creation Process

The basic steps involved in designing a codebook depend upon whether or not you have a survey typed with a word processor.

**Method 1:** If you do not have a word-processed survey, you are essentially "starting from scratch" and it will be necessary to manually enter the labeling for the codebook. Once completed, StatPac can automatically create a form for data entry and that can be loaded into your word-processor, an Internet survey, or an e-mail survey.

**Method 2:** If you already have a word-processed survey, considerable time can be saved by loading it into the Workspace window and then copying text from it to the codebook labels in the Variable Detail window.

### Method 1 - Create a Codebook from Scratch

There are three ways to set up a new codebook:

1. Use the codebook design features that are built into the program. The Grid and Variable Detail tools let you create and edit variables, as well as being able to extract variables from other studies or *libraries of questions*. A library of questions is simply a codebook with commonly asked questions. Rather than retyping a question with each new survey, you can extract it from a library.
2. Use Quick Codebook Creation (an Analysis utility program) to enter a format statement that describes the variables and their format. This is the fastest way to create a new codebook. However, the codebook it creates will not have any variable names, labels or value labels (although these can easily be added later).
3. If you import data from another format, a codebook will be created. Depending on the import format, the codebook may or may not have variable names.

## Method 2 – Create a Codebook from a Word-Processed Document

Save the survey with your word processor in .rtf (Rich Text Format). In StatPac, select File, Open, Rich Text File, and load the word-processed document into the workspace.

Activate the Variable Detail window by selecting View, Variable Detail, or by double clicking on “<New>” in the Variable List window. Then create the codebook one variable at a time by specifying a format for the variable, and copying selected text from the form to the Variable Detail window.

When creating a new variable, first type its format into the Variable Format field. Then copy text from the workspace to the Variable Detail window to fill in the rest of the variable information.

To copy text, first highlight the text on the form. It will automatically be copied to the clipboard when you highlight it. That is, it is not necessary to select Edit, Copy, or press <Ctrl C>. Next, click on one of the fields in the Variable Detail window. The text will be copied to the Variable Detail window. You can copy text from the form to the Variable Name, Variable Label, or Value Labels fields. Depending on the text, you may need to edit it in the Variable Detail window. This feature may be turned off by selecting Format, and then unchecking Semi-Automatic Copy/Paste.

---

## Multiple Response Variables

If an item on a questionnaire allows for more than one response, it is called a multiple response item. For instance, in the following question we would need to allow for five possible responses:

*Which of the following brands of toothpaste have you used in the last year? (Check all brands you've used)*

- ☐ *Gleem*
- ☐ *Colgate*
- ☐ *Pepsodent*
- ☐ *Crest*
- ☐ *Other*

Each of the five choices is viewed as a unique variable. That is, five variables would be required to accommodate all possible responses.

When designing a study in the Grid, using the Duplicate button will properly create all multiple response variables.

Generally, the following conventions are observed when creating multiple response variables.

1. The format for all multiple response variables must be the same.
2. The same (identical) variable label should be given to each of the multiple response variables.
3. If you will be creating a Web survey from the codebook, the number of variables must be the same as the number of value labels. Since there are five choices (value labels), there must be five identical variables.

The five variables for our example would contain the following information:

V1 Format:	N1
V1 Name:	Toothpaste
V1 Label:	Which of the following brands of toothpaste have you used in the last year?
V1 Value Labels:	1=Gleem 2=Colgate 3=Pepsodent 4=Crest 5=Other ;6

The second, third, fourth and fifth variables would be identical to the first variable, except the variable names would be: Toothpaste\_2, Toothpaste\_3, Toothpaste\_4, and Toothpaste\_5.

Note the above example uses value labels and a skip code. The skip code says to skip to variable six if nothing is entered for a variable.

The Missing OK parameter should be set to "Yes" for all five variables.

Note that during data entry, any toothpaste code can be entered for any variable. That is, if a person had only checked Crest, a "4" would be typed for the first variable. For the second variable, the data entry person would just press <enter> and this would cause the program to skip to variable six (the continuation of the questionnaire).

Sometimes surveys ask questions that limit the number of responses. For example, the following questionnaire item limits the respondent to two choices, even though there are five items listed. Note that the following method of limiting the number of choices may not be used for Web surveys.

*From the following list, choose the two items most important to you. (Two only please)*

\_\_\_\_\_ *Friendship*  
 \_\_\_\_\_ *Love*  
 \_\_\_\_\_ *Financial security*  
 \_\_\_\_\_ *Freedom*  
 \_\_\_\_\_ *Spirituality*

In this example, we controlled the number of responses by the way we asked the question. Two variables need to be created to hold the responses to this item (one for each check).

The study design would contain two variables for these multiple response variables:

V1 Format:	N1
V1 Name:	Important
V1 Label:	From the following list, chose the two items most important to you.
V1 Value Labels:	1=Friendship 2=Love 3=Financial security 4=Freedom

	5=Spirtuality ;3
V2 Format:	N1
V2 Name:	Important_2
V2 Label:	From the following list, chose the two items most important to you.
V2 Value Labels:	1=Friendship 2=Love 3=Financial security 4=Freedom 5=Spirtuality

Both items have the same format and variable label. Variable and value labels are only assigned to the first variable. The second variable will accept valid codes 1-5. Notice that the first variable also contains a skip pattern that says jump to variable three if nothing is specified for the first variable.

The two variables IMPORTANT and IMPORTANT\_2 are not weighted. That is, they could be swapped without affecting the results of any analysis (one is not more important than the other). Codes were assigned to each of the possible responses.

If the above question was asked in the following way, the variables would be weighted; that is, one variable is more important than the other:

*From the following list, write a 1 next to the item that is most important to you and a 2 next to the item that is second most important to you.*

\_\_\_\_\_ Friendship  
\_\_\_\_\_ Love  
\_\_\_\_\_ Financial security  
\_\_\_\_\_ Freedom  
\_\_\_\_\_ Spirituality

Notice that this is no longer a true multiple response question; it is really asking two different questions (which is first and which is second). Unlike the previous examples, both responses are not weighted equally. Whenever a question asks the respondent to rank a list of items in some sort of prioritized order, it is not multiple response. Instead, it is essentially a series of separate (but related) variables. Two variables would be created for this question, each having its own variable name, label and value labels:

V1 Format:	N1
V1 Name:	Most_Important
V1 Label:	From the following list, what item is the most important to you.
V1 Value Labels:	1=Friendship 2=Love 3=Financial security 4=Freedom 5=Spirtuality

V2 Format:	N1
V2 Name:	Second_Most_Impt
V2 Label:	From the following list, what item is the second most important to you.
V2 Value Labels:	1=Friendship 2=Love 3=Financial security 4=Freedom 5=Spirituality

While both variables in this example share the same value labels, they are still considered to be separate variables. The criteria to determine whether or not a question is multiple response is the issue of priority. If all responses are weighted equally, the question is appropriate for multiple response. If the question involves any sort of ranking of the items, it is best viewed as a series of individual variables.

When StatPac copies variables from the codebook to the data entry form, variables with the same variable label are interpreted as multiple response variables, and they will be automatically grouped together on the data entry template and in the HTML created for Web surveys.

---

## Missing Data

Missing data may be handled in one of two ways. Regardless of the method used, it is easy to change missing data using the Analysis program.

In most cases, no special provisions need to be made regarding what to do with missing data. If any variable in the data file is left blank, it will be treated as a missing value and will be excluded from the analysis. The analysis will print the number of missing cases, but will not include these when performing any statistical test.

The other method of handling missing cases is to enter an additional value label.

*A=6 years or less  
B=7 to 9 years  
C=10 to 12 years  
D=Over 12 years  
=Missing Cases*

Note that the code (on the left of the equals sign) is a space. All missing data in StatPac is stored as spaces (or blanks) during data entry.

When a variable is numeric, it is not appropriate to specify a value label of `<space>=Value Label`. Since a space is not a valid numeric code, it cannot be included in a numeric calculation. Therefore, missing data will automatically be excluded from most analyses of interval or ratio numeric data. It is possible, however, to recode missing data to a valid numeric value (such as zero), so that it will be included in the analyses. Also, several multivariate procedures include an option to use mean substitution for missing data.

It is important to understand the consequences of recoding numeric missing data to something else. Zero and missing are not the same. Analytical techniques involving computations on a variable treat zero differently than missing data. Missing data is

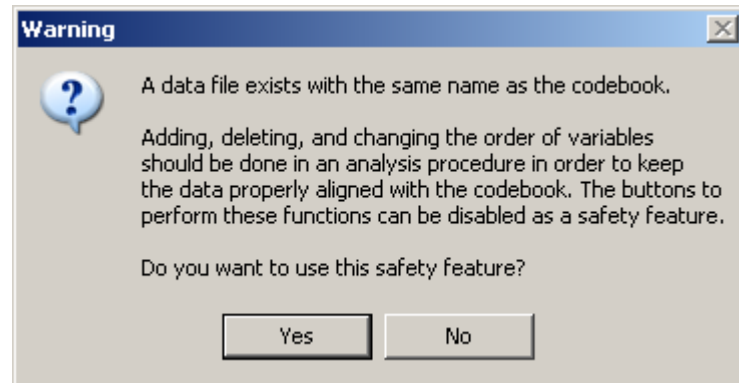
excluded from all numerical calculations, whereas zeros are treated just like any other numeric value.

---

## Changing Information in a Codebook

When initially designing a codebook and form, you can change any information for any variable. You can also insert new variables and delete existing variables. This will continue to be true up to the time that data is entered into a data file. After that, StatPac will issue a warning when you load a codebook that has an associated data file. StatPac gives this warning because these operations (i.e., adding new variables and deleting existing variables) would make the existing data file no longer match the codebook. You can, however, change any other study design information at any time.

If you receive the warning message, StatPac will let you activate a safety feature that prevents inadvertent additions or deletions of variables by disabling the Insert and Delete Buttons.



If you choose not to utilize the safety feature, be careful not to inadvertently add, delete, or change the order of any variables since this would make the existing data file incompatible with the modified codebook. However, you may still make changes to any other codebook information including a variable's format.

If you change the format of a variable, the associated data file adjusted accordingly. For example, if you change a variable format from A50 to A100, all the existing data records would be 50 characters too short. However, when you save the revised codebook, each data record will be padded with spaces so it matches the new codebook information. Note that this feature normally only changes one data file (the one with the same name as the codebook). Advanced users may wish to change multiple data files that all use the same codebook. To enable changing multiple data files, edit StatPac.ini and set *AllowMultipleDataFiles* = 1.

Advanced users may wish to turn on or turn off the safety feature so the prompt is not displayed. The *CodebookSafety* parameter can be edited in the StatPac.ini file to control this feature. Set *CodebookSafety* = 1 to always enable the safety feature, *CodebookSafety* = 2 to always disable the safety feature, and *CodebookSafety* = 0 (the default) to ask you each time that a codebook is loaded.

Note that the above information applies only when you load a codebook for which there is an associated data file.



This is important because entering a few records of dummy data is often the best way to discover errors in the study design. You would begin a typical project by designing the variables and creating a form. Then you could enter a few records into the data file as a test.

Entering a few dummy records is one of the best ways to test your codebook. You might discover a variable on the questionnaire that was inadvertently omitted from the study, an alpha field that's not wide enough to hold a response, or some other major change to the study design. If you don't need the data file (i.e., it's just dummy test data), you can simply delete the data file. To delete a data file, select File, Open, Data File. Right click on the data file you wish to delete and select Delete.

If you have already entered a substantial number of real data records, and then discover you need to add a new variable, you cannot simply add the variable to the codebook. Doing so would make the format of the codebook different than the data file. Instead, new variables should be created in an analysis, where both the codebook and the data file will be updated to include the new variable. See the NEW and SAVE commands.



# Data Manager Form

---

## Overview

The form is simply a template that can be used to enter and edit data. It can be created automatically by StatPac, or you can create it manually (with StatPac or your word processor). To create an automatic data manager form, first load the codebook and then select Design, Data Manager Form,

There are two components to the form. One is the text itself. The text is simply the questions on the survey. The text can be typed directly into the Form window or loaded from an existing word-processed document. The other component is the *data input fields*. The data input fields define where the answers will be typed on the form. The form is a combination of text and data input fields.

---

## Data Input Fields

Data input fields will be shown on the form in another color (making them easy to identify). They will appear as a variable number enclosed in brackets. For example, variable twelve might look like this [12]. Generally, the form will have one data input field for each variable in the codebook.

Data input fields can be inserted or deleted from the form during the study design process.

To insert a data input field on the form, first select the desired variable from the Variable List or Variable Detail window. Then hold the Alt key and click the left mouse button where you want the field to be located on the form. The current variable will be incremented each time you insert a data input field.

To delete a data input field from the form, highlight it on the form and click on the Cut Button, or select Edit, Cut (or use the Ctrl X shortcut). You can highlight multiple data input fields delete them all at once.

Data input fields will be automatically placed on the form when you copy variables from the codebook to the form.

---

## Form Naming Conventions

In most cases, the codebook, form, and data file will share the same name. A codebook called RESEARCH would have an associated form called RESEARCH,

and you would probably enter the data into a data file called RESEARCH. You will use this simple naming scheme for nearly all studies.

However, there are situations where you may want to use different names for the codebook, form, and data file. The form is simply a template for displaying the data. A form can be used to display all the variables, or just some of them. You can have several different forms for a given codebook. Each form would have its own name, and each could show (or not show) any of the variables. Thus, different forms could be used to give different "views" of the same data

---

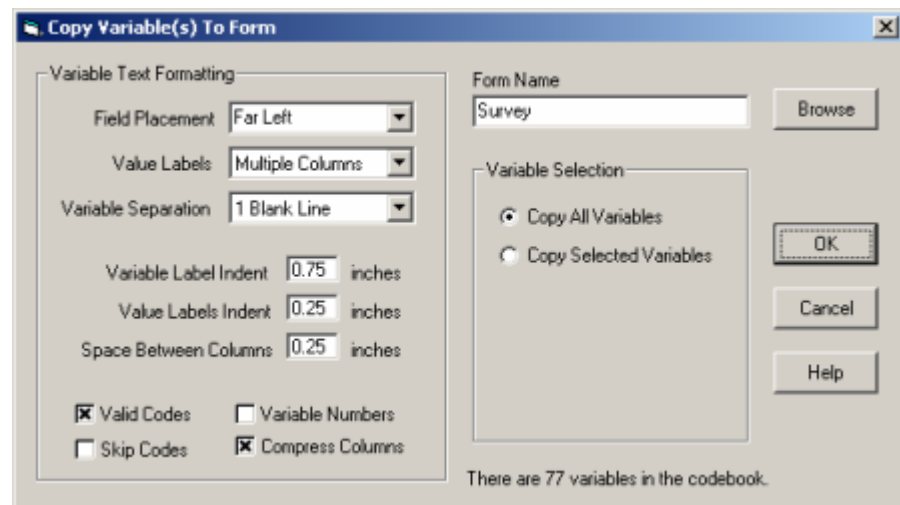
## Form Creation Process

There are two basic ways to create a form. One is to use the codebook to automatically create the form, and the other is to use text from a word-processed file as the foundation for the form. In nearly all cases, you will use the codebook to create the form and then you can modify it as necessary.

### Using the Codebook to Create a Form

The form is created after the codebook has been completed.

Automatic form creation involves copying variables from the codebook to the form, and formatting them according to your specifications. To create an automatic form, first load the codebook. Then copy all the variables from the codebook to the form by selecting Design, Copy Variable(s) To Form. The Variable Text Formatting controls how the variables will be formatted on the form.

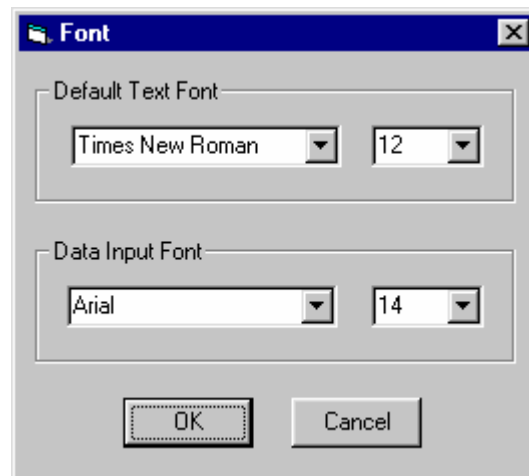


If you want to delete some or all of the variables from the form, highlight the text you want to delete and click the Cut Button (or choose Edit, Delete) to delete the variables from the form. To delete all the variables on a form (for example, to just start over), choose Edit, Select All, and then click the Cut Button or choose Edit, Delete. **The codebook is not affected by any changes made to the form.**

If you inadvertently delete an input field or entire variable it can be easily re-inserted. To insert only a data input field on the form, first select the desired variable from the Variable List or Variable Detail window. Then hold the Alt key and click

where you want the field to be located on the form. The current variable will be incremented each time you insert a data input field. To insert a data input field and the text for the variable, hold the Ctrl key and click where you want to insert it on the form.

The font name and font size that will be used to create the formatted variable can be set by selecting Format, Set Default Fonts. The Default Text Font will be used to create the text on the form. The Data Input Font will be used during data entry to enter the data into the fields.



## Using a Word-Processed Document to Create a Form

If you already have typed your survey with a word-processor, you can use it as the foundation of the form. You probably already used it to expedite the creation of the codebook.

First, using your word-processor, save the survey in Rich Text Format (.rtf).

Next you need to create a blank form. Load the codebook and the select Design, Data Manager Form. Click Copy Selected Variables and type 0 for the Variables to Copy. Then click OK. A blank form will be created.

Open the rich text file in the Workspace window. Select Edit, Select All, Edit Copy (or type Ctrl A, Ctrl C). Open the blank form and Select Edit, Paste (or type Ctrl V). The text from the rich text document will now be part of the Data Manager form.

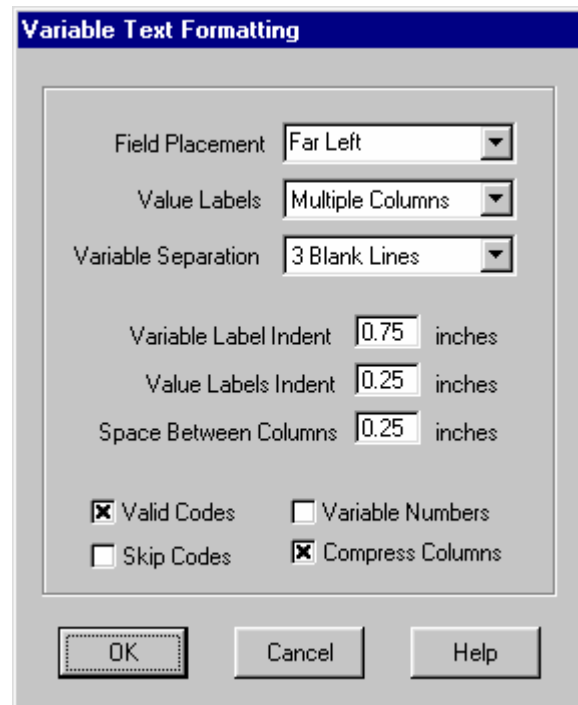
The final step is to insert the data input fields on the form. These must be added manually. To insert a data input field on the form, first select the desired variable from the Variable List or Variable Detail window. Then hold the Alt key and click the mouse where you want the data input field to be located on the form. The current variable will be incremented each time you insert a data input field. Continue until you have a data input field for all variables on the form.

If you click the mouse in the wrong place, use the Cut Button to delete the data input field, and the Paste Button to insert it at the correct location. A data input field cannot be deleted with the Delete or Backspace keys. It can be deleted with the Cut Button or by selecting Edit, Cut.

---

## Variable Text Formatting

*Automatic variable creation* is an important part of both manual and automatic form creation. The purpose of automatic variable creation is to reduce typing. StatPac will allow you to selectively transfer information from the codebook to the form, and it gives you the ability to automatically format this information in a variety of ways. The Variable Text Formatting Dialog window lets you adjust the parameters that control the format for this information on the form. Select Options, Variable Text Formatting to modify the formatting specifications.



Each component of the Variable Text Formatting Dialog window can be modified.

### Field Placement

Field placement refers to the location for the data input field (i.e., where you want the cursor located when you're ready to enter data for the variable). Data input fields will be displayed in a different color on the form. There are six possible field placements: left, far left, right, far right, below, and far below.

#### ***Field placement set to Left***

**[1]** 1. How would you rate your expectation for this seminar?

#### ***Field placement set to Far Left***

**[1]** 1. How would you rate your expectation for this seminar?

### **Field placement set to Right**

1. How would you rate your expectation for this seminar? [1]

### **Field placement set to Far Right**

1. How would you rate your expectation for this seminar? [1]

### **Field placement set to Below:**

1. How would you rate your expectation for this seminar?

[1]

1= High      2=Medium      3=Low

### **Field placement set to Far Below:**

1. How would you rate your expectation for this seminar?

1= High      2=Medium      3=Low

[1]

## **Value Labels**

The Value Labels parameter lets you format the value labels in single or multiple columns.

### **Value labels set to Single Column**

A=Low

B=Medium

C=High

### **Value labels set to Multiple Columns**

A=Low    B=Medium    C=High

## **Variable Separation**

The Variable Separation parameter controls the spacing between variables. It is especially useful when copying multiple variables from the codebook to the form. The parameter can be set to blank line(s), a page feed, or a horizontal line.

If you want to create a form with one variable per page, you would use a page feed as the variable separator, and set the View mode to Page View before saving the form. During data entry, each variable will be displayed on its own page.

## **Variable Label Indent**

The variable label indent refers to the number of inches that the variable label will be indented from the left margin. This is especially useful when the field placement is set to Far Left

**Example: Variable label indent set to ½ inch**

[1] 1. How would you rate your expectation for this seminar?

**Example: Variable label indent set to 1 inch**

[1] 1. How would you rate your expectation for this seminar?

## Value Labels Indent

The value labels indent refers to the number of inches that the first column of value labels will be indented, with reference to the position of the variable label. The following examples will illustrate the value labels indent:

**Example: Value label indent set to 0 inches**

[1] 1. How would you rate your expectation for this seminar?  
A=Low  
B=Medium  
C=High

**Example: Value label indent set to ½ inch**

[1] 1. How would you rate your expectation for this seminar?  
A=Low  
B=Medium  
C=High

## Space between Columns

The space between columns refers to the number of inches that will be used to separate the columns of value labels. This parameter only applies when the Value Labels parameter is set to multiple columns. The actual space required for a column is determined by the lengths of the value labels themselves.

**Example: Space between columns set to ¼ inch**

[1] 1. How would you rate your expectation for this seminar?  
A=Low B=Medium C=High

**Example: Space between columns set to ½ inch**

[1] 1. How would you rate your expectation for this seminar?  
A=Low B=Medium C=High



## Valid Codes

When the Valid Codes parameter is set, the valid codes will be included with the value labels. This specifically refers to those valid codes that are not part of a <Code>=<Label> line.

### ***Example: Valid codes is set***

[1] 1. How would you rate your expectation for this seminar?  
1-5            0=Don't know

## Skip Codes

When the Skip Codes parameter is set, all skip codes will appear as part of the value labels. Generally, skip codes would not be shown as part of the data entry form since the branching will occur automatically.

## Variable Numbers

When the Variable Number parameter is set, the variable number will be included as part of the variable label. When included, it will precede the variable label and have a period terminator. It is important to note that the variable number might not be the same as the item number on the survey. Surveys often contain multiple response items or groups of "sub-questions" under the same "item number" on the survey.

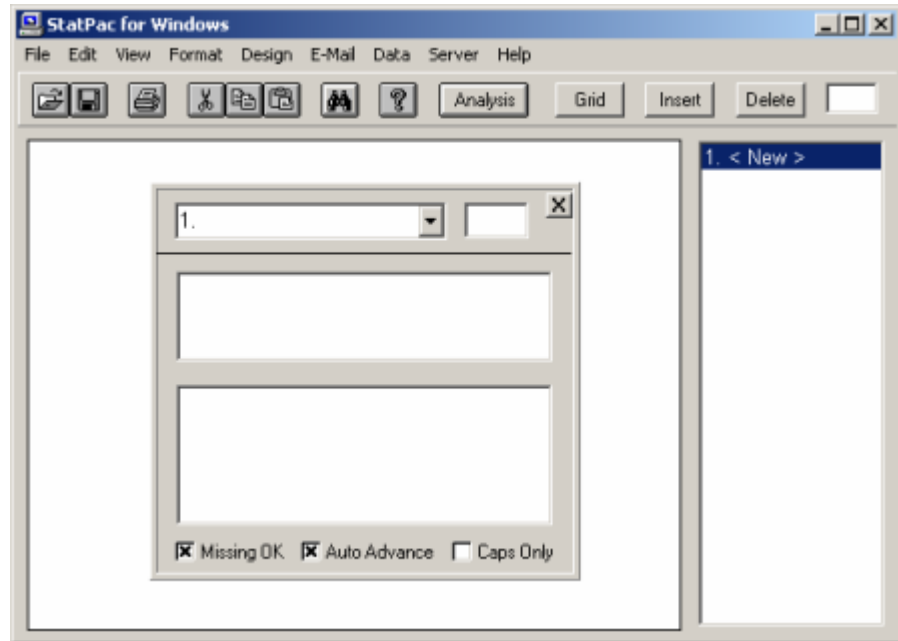
---

# Variable List and Detail Windows

The Variable List window allows the user to view the variable names and variable numbers while entering data. Clicking on a variable in the Variable List window will make that variable the current variable. Double clicking on a variable in the Variable List window will activate the Variable Detail window.

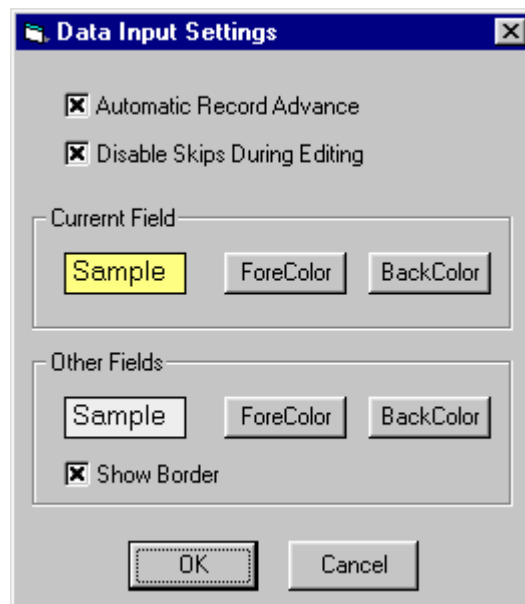
The Variable Detail window gives complete information on the current variable. It can be turned on or off by selecting View, Variable Detail. Double clicking on a field will also activate the Variable Detail window.

The Variable Detail window can be dragged around the screen by clicking and holding the left mouse button on any gray area in the Variable Detail window.



## Data Input Settings

You can set the default data input settings. These can be changed during data entry. Select Options, Data Input Settings to change the settings.



The Automatic Record Advance option controls whether the current record will be automatically incremented when data entry person reaches the last field in the current record.

If the Automatic Record Advance option is set, the program will automatically advance to the next record when the last field of the current record has been entered. This way, the data entry person will be able to enter a large number of records without clicking on the New Record Button for each record.

The Disable Skips During Editing option controls whether skip codes should be active when editing an existing record. When this option is set, skip codes will work when entering a new record and will not work when editing an existing record.

The final Data Input Settings let you change the colors for the fields. During the Study Design, all data input fields will be shown with the "Current Field" colors. During data entry, only the current data input field will use the "Current Field" colors, and the other fields will be displayed with the "Other Field" colors.

---

## Select a Specific Variable

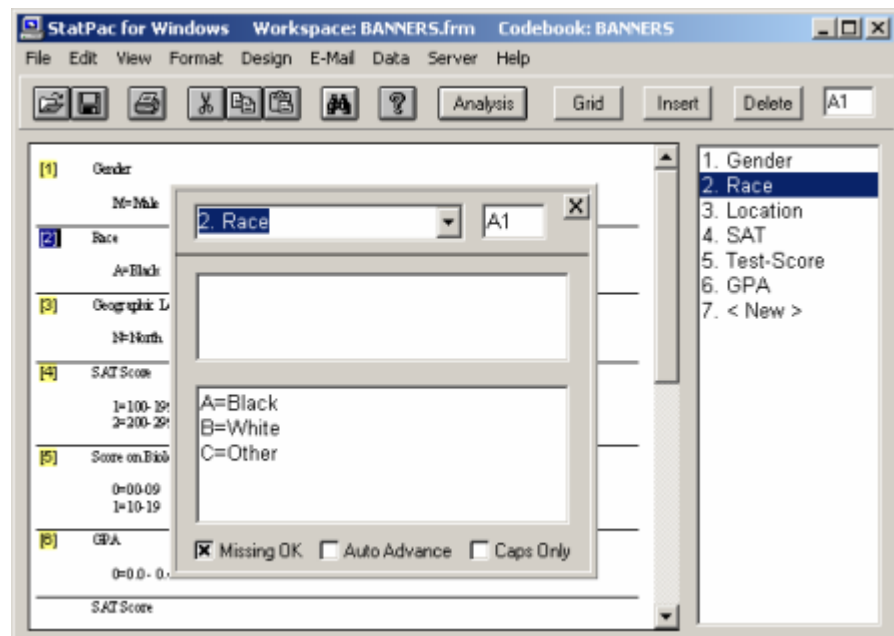
There are three ways to select a specific variable:

When the Variable List Window is displayed, you can select a variable by clicking on it. Double clicking on a variable will also display the Variable Detail window.

When the Variable Detail window is displayed, you can select a variable by selecting it from the variable name field.

If the form already contains an input field for a variable, you can select the variable by clicking on the input field. Double clicking on an input field will also display the Variable Detail window.

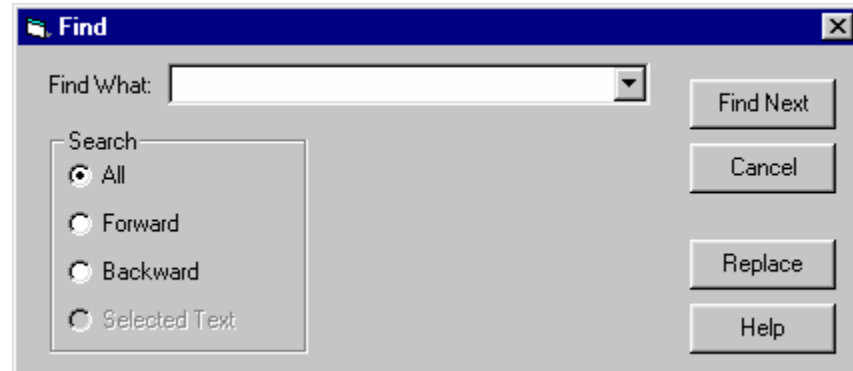
The three highlighted areas show the three places you can click to select a variable.



---

## Finding Text in the Form

Use the Find Dialog window to search for specific text in the form. Select Edit, Find (or use the Ctrl F shortcut) to display the Find Dialog window.

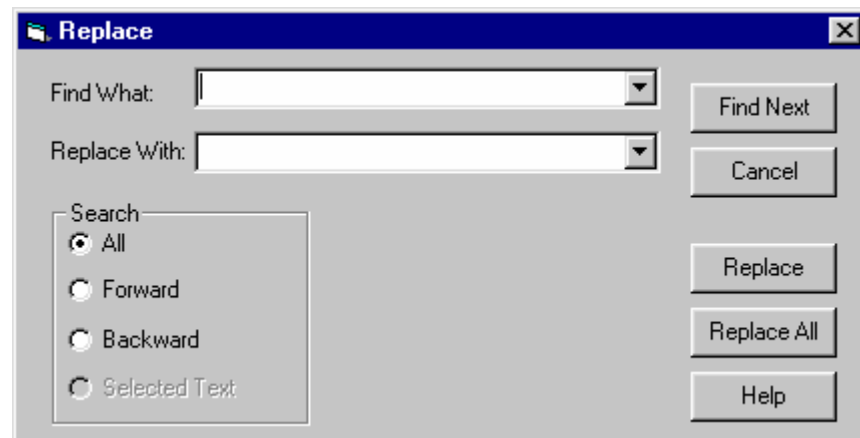


To begin a search, type the search text and click on the Find Next Button. After a search has been started and a match has been found, you can continue the search by clicking on the Find Next Button. Upper and lower case differences will be ignored in the search.

---

## Replacing Text in the Form

Use the Replace Dialog window to replace specified text in the form. Select Edit, Replace (or use the Ctrl H shortcut) to display the Replace Dialog window. Alternatively, you can click the Replace Button from the Find Dialog window.



Upper and lower case differences will be ignored when finding text. However, replaced text will use the exact text typed into the Replace With window.

---

## Saving the Codebook or Workspace

Select File, Save Codebook or Save Workspace to save the codebook or Workspace window. **It is recommended that you save your work at regular intervals.** You may exit from a partially completed codebook or form and finish them at another time. Click the Save icon (a picture of a diskette) to save both the codebook and the Workspace window.



# Data Manager

---

## Overview

The Data Manager is used to enter new data, edit existing data, verify data using a double entry method, and print data using a form created with the Study Design program. All of the buttons on the opening screen will be disabled until a data file has been loaded.

In order to use the Data Manager, you must have first created a data manager form. It is not possible to enter new data or edit existing data without a form.

The form provides the data entry template (i.e., what the data entry person will see on the screen). The Data Manager will attempt to show the current field near the middle of the screen. This means that during data entry, the cursor will appear to remain relatively stationary near the middle of the screen, and the form will scroll after each field is entered.

The screenshot shows a Windows-style window titled "Data File: ATTRIBUTES". The menu bar includes "File", "Edit", "View", "Options", and "Help". Below the menu bar is a toolbar with buttons for "New Record", "Record Select" (with left and right arrows), "Field Select" (with up and down arrows), and "DUP" and "DEL" buttons. The main area of the window displays a form titled "Attributes Study" in a large, bold, underlined font. Below the title, there is a line of text: "The form is just a word-processed document with data input fields." followed by a line of text in a cyan box: "You can make it look any way you want." Below this, there is a list of six questions, each preceded by a number in a box: "1 How would you rate the taste?", "2 How would you rate the smell?", "3 How would you rate the texture?", "2 How would you rate the appearance?", "1 How would you rate the package design?", and "2 What is your overall rating for this product?". To the right of the list, there is a legend: "1=Excellent", "2=Good", "3=Fair", and "4=Poor".

There are two ways to run the Data Manager. If the data file already exists, and you want to add, edit, or delete records, select File, Open, Data File. If a data file does yet exist, you can select Data, Run Data Manager.

---

## Keyboard and Mouse Functions

The following are the basic keyboard functions for record and field selections:

Enter	Advance to the next field using skip pattern if present
Tab	Advance to the next field using skip pattern if present
Shift Tab	Move to the previous field. Does not follow skip patterns.
Down Arrow	Advance to the next field using skip pattern if present
Up Arrow	Move to the previous field. Follows skip patterns that were used.
Page Up	Move to previous record.
Page Down	Advance to next record.

---

## Create a New Data File

To create a new data file, first run the Data Manager. Then select File, Open, and type a new file name. In nearly all cases, the data file name will be the same as the codebook and form names. The program will first ask for the name of the form you want to use as your data entry template. Then it will ask for the name of the data file. The form will load and you will be able to begin entering data.

---

## Edit or Add To an Existing Data File

To edit or add records to an existing data file, first run the Data Manager. Then select File, Open and select the data file. If there is a form with the same name (most situations), it will be loaded and you will be able to edit or enter data. If there is not a form with the same name, you will be asked to enter the name of the form.

---

## Select a Different Data File

It is not necessary to close the current data file before choosing to edit a different data file. To change data files select File, Open and select the data file. The current data file will be closed automatically before the program opens the new file.

---

## Change Fields

The current field is the one you are ready to enter data into. It will be highlighted on the form using the color scheme developed during the design of the form (although the colors can be changed with the Data Manager program).



During normal data entry, the current field will change automatically following skip patterns if necessary. The user, however, is free to manually change to any field at any time. There are many ways to change fields.

Clicking the mouse on any field will make it the current field. Clicking the mouse on the Previous Field Button or the Next Field Button will change to the previous or next field. You can also type a field number into the Current Field window and press enter.

If the Variable List window is displayed, clicking on a variable will make that variable the current field. If the Variable Detail window is displayed, selecting a new variable will make it the current field. The Variable List and Variable Detail windows can be displayed by selecting View.

Finally, the following keys can also be used to manually change fields.

Enter	Advance to the next field using skip pattern if present
Tab	Advance to the next field using skip pattern if present
Shift Tab	Move to the previous field. Does not follow skip patterns that were used.
Down Arrow	Advance to the next field using skip pattern if present
Up Arrow	Move to the previous field. Follows skip patterns that were used

---

## Change Records

The Automatic Record Advance option controls whether the current record will be automatically incremented when data entry person reaches the last field in the current record.

If the Automatic Record Advance option is set, the program will automatically advance to the next record when the last field of the current record has been entered. This way, it is possible to enter a large number of records without clicking on the New Record Button for each record. To change the Automatic Record Advance option, select Options, Data Input Settings.

There are several ways to manually change records. Clicking on the Previous Record Button or the Next Record Button will change to the previous or next record. You can also type a record number into the Current Record window and press enter. Finally, the following keys can also be used to manually change records.

Page Up	Move to previous record.
Page Down	Advance to next record

---

## Enter a New Data Record

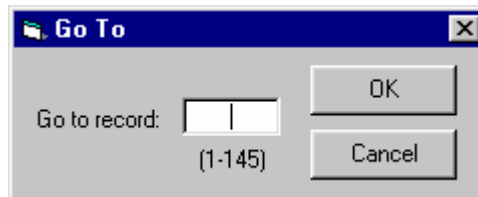
There are two ways to enter a new data record. The first is to click on the New Record Button, and the second is to select Edit, Begin New Record. The current

record number window will be updated to show the record number for the new record. If you do not enter any data for the new record, it will not be saved. If any data is entered, the new record will be added to the end of the data file. It is not possible to insert a new record into the middle of an existing data file.

---

## View Data for a Specified Record Number

There are two ways to view the data for a record with a known record number. The first is to type the desired record number into the current record number window and press enter (or click anywhere on the form). The second is to select Edit, Go To Record (or use the Control G shortcut).

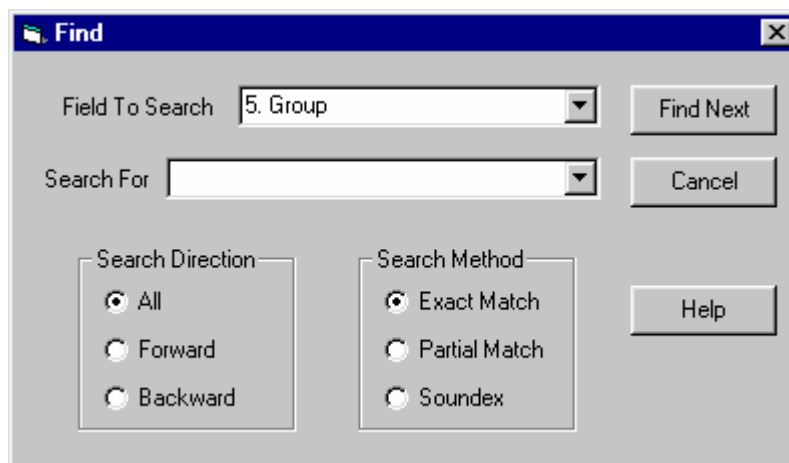


When finished, the selected record will be displayed, and the focus will remain on the currently selected field. The current record number window will be updated to show the record number for the selected record.

---

## Find Records That Contain Specified Data

StatPac makes it easy to find data records that contain specified data. To begin a search, click the Find Button, or select Edit, Search Data File (or use the Control F shortcut). The Find dialog window will be displayed.



After a search has been started and a match has been found, you can continue the search by selecting Edit, Continue Search (or use the [F3] shortcut key). The Find dialog window does not have to be displayed to continue a previous search, although if it is displayed, you can also click on Find Next to begin the search. Note that the search often happens so quickly that it seems instantaneous. The user can watch the current record number to confirm the record number that they are viewing following a search.

There are four components in the Find dialog window.

## Field To Search

Select the variable (i.e., field) you want to search. The default will be the current variable.

## Search For

This is the text or value you want to search for. If you want to search for missing data, leave this field blank.

## Search Direction

Select All, Forward, or Backwards.

When the Search Direction is set to All, the search will begin with the record following the current record (e.g., if record 5 is displayed on the screen, the search will start with record 6). If no match is found by the end of the file, the search will continue with record 1 and continue until all records have been examined. If no match is found, StatPac will report it, and the current record will continue to be displayed. When the Search Direction is set to Forward or Backward, the search will go only to the end or beginning of the file (respectively).

## Search Method

Select Exact Match, Partial Match or Soundex.

Upper and lower case differences will be ignored for all searches regardless of the method used. If you search for JOHN JONES, John Jones will be found.

If you set the Search Method to an Exact Match, then the data must exactly match the search string (with the exception of case differences). If you search for Jones, StatPac will not find a record that contains John Jones in the search field

If you set the Search Method to a Partial Match, then the search will find records that contain the search string regardless of other text in the field. If you search for Jones, StatPac will find records that contain John Jones.

A Soundex search is one that uses the sound of the word instead of its exact spelling. This makes it possible to find text even when there are spelling or data entry errors. If you set the Search Method to a Soundex Search, then the search will find records that contain the sound of the search string regardless of other text in the field. If you search for Jonathan, StatPac will find records that contain John Jonethon, Fred Johnathon, Mary Ann Jonathon, etc.

---

## Duplicate a Field from the Previous Record

It is sometimes desirable to be able to repeat data from the previous record. For example, assume one of the variables in your study is the full city name. During data entry you would type the city name for the first record. When entering data for the city field in the second (and subsequent) records, you could duplicate the response from the previous record. The duplicate field function is disabled during double entry verification.

There are two ways to duplicate the data from the previous record. The first is to click on the DUP Button, and the second is to select Edit, Duplicate Field From Previous Record (or use the [F5] shortcut key). The duplicate function will copy the contents of the previous record to the current record for the current field.

---

## Delete a Record

Deleting a record is a fairly common procedure because duplicate records are often discovered in a data file. There are two ways to delete a record. The first is to click on the DEL Button and the second is to select Edit, Delete Current Record.

StatPac will not actually delete the record at this time. Instead, the contents of all variables in the record are set to blanks (missing). In this way, the record is marked for deletion rather than actually being deleted. The rationale being that deleting a record would cause all subsequent records to move down in the data file, in effect changing their record numbers. Since editing is often done by record number, it is important that the record numbers do not change during an editing session.

Records marked for deletion cannot automatically be undeleted. When you mark a record for deletion, it is actually stored as blanks in the data file, so there is no way to recover the information. You can, however, re-enter the data for that record. If you enter any data in a record marked for deletion, it will not be deleted since it now contains data.

When you exit the Data Manager, you will have the opportunity to compact the file (i.e., eliminate the deleted records). This will change the record numbers for future editing sessions. If you want to preserve the record numbers for the next editing session, don't compact the file.

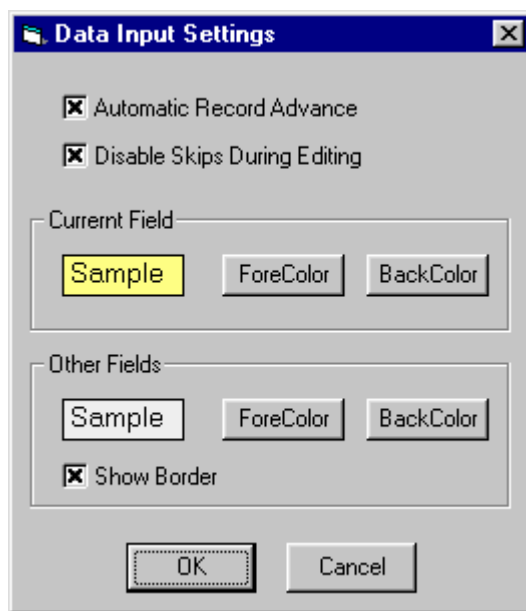
You can also manually compact the data file at any time by selecting Options, Compact Data File. You will be asked to confirm your selection because the procedure will change the record numbers. The delete record function is disabled during double entry verification.

If you have deleted records, it is important to compact the data file before performing any statistical analysis. Otherwise, the deleted records would be counted as missing data in the analysis.

---

## Data Input Settings

Most data entry control parameters are specified in the Study Design program. However, a few parameters can be altered by the data entry person. Select Options, Data Input Settings to change the settings.



The Automatic Record Advance option controls whether the current record will be automatically incremented when data entry person reaches the last field in the current record.

If the Automatic Record Advance option is set, the program will automatically advance to the next record when the last field of the current record has been entered. This way, it is possible to enter a large number of records without clicking on the New Record button for each record.

The Disable Skips During Editing option controls whether skip codes should be active when you are editing an existing record. When this option is set, skip codes will work when you are entering a new record and will not work when editing an existing record. If you are entering a new data record, and temporarily return to a previous record, and then back to the new record you were working on, then you will no longer be entering a “new” data record. In other words, a record becomes permanent as soon as you change records (even if only some of the information was entered for that record). Skip codes will be disabled when you return to the “new” record that you were entering unless the Disable Skips During Editing is unchecked.

The final Data Input Settings let you change the colors for the fields.

---

## Compact Data File

When you delete a record with the Data Manager, StatPac makes it a blank record without actually deleting the physical record from the data file. If you have deleted records, it is important to compact the file by deleting the blank records. Otherwise, the deleted records would be counted as missing data in the analysis. To compress a file, select Options, Compact Data File. See Deleting A Record. The compact data file function is disabled during double entry verification.

---

## Double Entry Verification

StatPac performs validity and range checking on all data entered. However, some people prefer to use a Double Entry Verification method to further reduce data entry error. Using this method, all the data is first entered into a data file. Then the data is entered again and compared against the first data during the second data entry process. Discrepancies are brought to the attention of the data entry person and they are resolved in real time during data entry.

To begin Double Entry Verification open the data file to be verified and select Options, Double Entry Verification. All the existing data will be "hidden" from the user, and it will appear that you are ready to begin entering data beginning with the first record. As each field is entered it is compared with the existing data. Skip patterns will be followed as if this were the same as entering a new data record. Discrepancies will be highlighted and you will be able to specify whether to use the new data or original data. Once a field has been verified, its data will no longer be hidden.

Obviously, Double Entry Verification works only if surveys are entered in exactly the same order as the original data entry. When operating in the Double Entry Verification mode, each time a new record is displayed, the field data will appear blank even if that record was already verified. Thus, if the data entry person stops midway through the verification process, it is important they note the record number they were working on when they stopped, so they can begin at that record when they return.

---

## Print a Data Record

To print a data record select File, Print. The current record will be printed using the form as a template.

---

## Variable List & Detail Windows

The Variable List window allows the user to view the variable names and variable numbers while entering data. Clicking on a variable in the Variable List window will make that variable the current variable. Double clicking on a variable in the Variable List window will activate the Variable Detail window.

The Variable Detail window gives complete information on the current variable. It can be turned on or off by selecting View, Variable Detail. Double clicking on a field will also activate the Variable Detail window.

The Variable Detail window can be dragged around the screen by clicking and holding the left mouse button on any gray area in the Variable Detail window.

---

## Data File Format

StatPac stores its data files in sequential ASCII fixed format with a carriage return and line feed at the end of each record. This is sometimes referred to as a flat ASCII file. The data file also contains an end of file mark (*Ctrl Z or ASCII 26*) at the end of the file. This is the most universally accepted data file format, and many data base managers refer to it as SDF format

When using data created by another program, you can determine if it is an ASCII file by loading into a text editor. If the data does not appear as garbage, it's probably ASCII. If all records appear to contain the same number of characters, it's probably fixed format. Data in any other format must be imported into StatPac.

A "record" or "case" in StatPac is defined as a string of characters terminated with a carriage return and line feed. Fixed format means that all records are exactly the same length (i.e., they contain the same number of characters) as all other records.

Two different fixed format data files are shown as follows:

	<u>Single variable file</u>	<u>Multiple variable file</u>
Record 1	14	14A3184172
Record 2	29	29C2018061
Record 3	06	06B9012103

and so on ....

Notice that a data file is just a series of numbers (or letters). All the records are always the same length. In this example, the single variable file uses two columns per record and the multiple variable file uses ten columns per record (with an unknown number of variables). The end of each data record contains an invisible carriage return and line feed, so each record actually contains two more characters per record (one for the carriage return and one for the line feed).

If you already have a data file stored in sequential ASCII format, you can use it with StatPac by giving it a .dat extension. You only need to set up a codebook to match the format of the data file. No other changes are necessary.

Most data downloaded from a mainframe, read-in from a scanner, or received from a data entry company will be in sequential ASCII format. To use this data with StatPac, perform the following three steps:

1. Set up a study design with StatPac that exactly matches the format of the data file. The format of the study design must exactly match the format of the data you will be using. If the data file contains unused columns, dummy variables should be created in StatPac to "pass-over" the blank portions of the data record.
2. Copy the data file to your work subdirectory. Then rename it to the same name as the codebook except with a .dat extension.

If you need to use a data file that is not stored in sequential ASCII format, you must import it into StatPac. The Analysis program has a utility to import most foreign data files into StatPac's format. This makes it easy to exchange data with other software packages, including most data base managers and spreadsheets.





# Email Surveys

---

## Overview

When most people refer to an *email survey*, they mean an email invitation with a embedded link to an Internet survey. If this is what you want, then refer to the Internet survey section and Email List Management sections of this manual.

This chapter is for users who want to send emails, where the email itself is the survey.

An email survey is one where the body of the email contains the survey itself. This is in contrast to an Internet survey where you send an email invitation and the body of the email contains a link to the Internet survey.

The problem with email surveys is that you don't know what kind of email reader your respondents are using. A survey that looks fine in one email reader might appear distorted in another email reader. Therefore, we discourage you from sending email surveys.

Nevertheless, there are two kinds of email surveys: HTML and plain text.

---

## HTML Email Surveys

In an HTML Email survey, the email you send is the survey itself. When the respondent views the email, they will see the survey.

Follow the same procedure as you would if creating an Internet survey, including uploading the survey to the Internet. Then, instead of sending the respondents a link to the survey, you send them the first page of the survey (i.e., the `_1.htm` file).

When they open the email, they will see the first page of the survey, and when they click the Submit button, the responses will be stored on the server and the second page or thank you page will be displayed. Data is retrieved from the server as if it were an Internet survey.

Conceptually, HTML email surveys are wonderful. However, in reality, HTML email surveys can cause several problems. Some people have their email readers set to not display HTML. Other people have their email readers set to disallow executables which means they delete the JavaScript in the HTML that StatPac relies on. This will cause them to see a JavaScript error when they view the email. Additionally, you cannot use cookies or ID numbers to track who responded and who didn't.

For these reasons, we discourage you from sending HTML email surveys. If you choose to send them anyhow, please do so knowing that some respondents are likely to have problems.

---

## Plain Text Email Surveys

A plain-text Email is one where you create a survey using ASCII text and include it as part of the body text in a regular Email. The survey is emailed to a potential respondent. They click their Reply button, complete the survey, and then click the Send button to return the completed survey to you. Upon receipt of their reply, you can immediately import it into a StatPac data file, or filter it to a mailbox (i.e., file) with other returned Emails and import all the responses at once. Importing returned Emails appends the respondent's answers to the end of a StatPac data file.

There are two kinds of variables that can be included in a plain-text Email. One is where the respondent uses an X to check the appropriate response(s). The other is where the respondent fills in a text or numerical response. The following simple survey shows both kinds of variables. The first five survey items are "X the box" and the last five items are "fill in the blank":

TO PARTICIPATE, PLEASE CLICK THE REPLY BUTTON.

Type an X between the brackets to indicate your selection.

1. Gender   ☐ Male   ☐ Female

2. Ethnicity:

☐ Black

☐ White

☐ Other

3. What is your favorite brand?

☐ Brand X     ☐ Brand Y

☐ Brand Z     ☐ Undecided

4. Overall, how would you rate our product?

(Type an X between the brackets to indicate your response.)

Excellent

Poor

☐ ☐ ☐ ☐ ☐ ☐ ☐

1   2   3   4   5   6   7

5. Where did you hear about the product?

(Type an X between all brackets that apply.)

☐ Radio

☐ TV

☐ Newspaper

Excellent                      Poor

1 2 3 4 5 6 7

Type the number of your rating here. [ ]

8. What do you feel would be a fair price for this product?  
[       ] Type the dollar amount between the brackets..

10. What could we do to make our product better?  
(Type your answer between the brackets).  
[

CLICK THE SEND BUTTON TO FINISH THIS SURVEY

## Markets

Notice that the brackets are used to show respondents where to type their answers. The text of the Email is completely free-form, except that brackets may only be used to specify where respondents are supposed to type their answer. You may not use brackets anywhere else in the Email.

The numbering of the survey items is optional. The purpose of these numbers is to guide the respondent from one question to the next. They are not necessarily the same as the variable numbers in the codebook. That is, some survey items might be multiple response and occupy multiple variables in the codebook. For example, survey item 5 is multiple response and takes three codebook variables. From that point on, the item numbering on the survey is no longer is the same as the codebook variables.

Numbering the items on an Email survey also serves an additional purpose. When a respondent replies to your Email, StatPac will extract their answers from the Email by looking at what's between the brackets. If a respondent inadvertently deleted one of the brackets, StatPac might get confused on which variable it was evaluating. When item numbering is not used, StatPac will report the error and not add any of that respondent's answers to the data file. When item numbering is used, StatPac will

be able to identify the start of a new variable or series of multiple response variables, and it will leave only the defective item blank. StatPac will report the error, but the rest of the data for that respondent will be added to the data file. Thus, item numbering allows StatPac to "re-synch" with the variables in the codebook, even when a bracket has been deleted.

If you do use item numbering on your Email survey, the number may begin with the letter Q (a common abbreviation for question). It must also end in a period followed by a space, and it may not contain non-numeric characters. Each of the following would be correct usage of item numbering.

5. Where did you hear about the product?

Q5. Where did you hear about the product?

Q5. (a) Where did you hear about the product?

The following would be incorrect usage of item numbering. The first example does not have a period following the number. The second example has a space after the Q. The third example has the letter "a" before the period and space.

5 Where did you hear about the product?

Q 5. Where did you hear about the product?

Q5a. Where did you hear about the product?

## Codebook Design for a Plain Text Email Survey

The codebook for an Email survey is the same as any other codebook. The rule is, if you want the respondent to use an X to show her response, the codebook must specify a value label for each response. If you want the respondent to fill in an answer, do not specify any value labels for that variable.

Variables can be alpha or numeric. When multiple sets of brackets (boxes) are required, they can be placed horizontally (items one and four) or vertically (items two and five). If you specify more than one column of boxes (item three), they must be positioned horizontally. That is, the first box is associated with value label 1 and the box to the right of it is associated with value label 2. In the second row, the first box is associated with value label 3, and the box to the right of it is associated with value label 4.

Here is the codebook for the first five items:

V1. Gender (A1)

M=Male

F=Female

V2. Ethnicity (N1)

1=Black

2=White

3=Other

V3. What is your favorite brand? (N1)

1=Brand X

2=Brand Y

3=Brand Z  
 4=Undecided  
 V4. Overall, how would you rate our product? (N1)  
 1=Excellent  
 2=  
 3=  
 4=  
 5=  
 6=  
 7=Poor  
 V5. Where did you hear about the product? (N1)  
 1=Radio  
 2=TV  
 3=Newspaper  
 V6. Where did you hear about the product? (N1)  
 1=Radio  
 2=TV  
 3=Newspaper  
 V7. Where did you hear about the product? (N1)  
 1=Radio  
 2=TV  
 3=Newspaper

The last five items on the sample survey are fill-in-the-blank items. The variable formats in the codebook specify the maximum response length for each of the items. The actual distance between the opening and closing bracket on the Email survey can be any length. If the respondent "stretches" the space between the brackets by typing a longer response, the data will be recorded properly in the data file provided the answer does not exceed the field length specified in the codebook. If the respondent's answer does exceed the field length, StatPac will allow you to adjust the codebook so that all fields are sufficient to hold the responses. Also note that the opening and closing brackets do not need to be on the same line of the Email. Item ten in the example offers the respondent several lines for their answer. Here is the codebook for the second five items

V8. Overall, how would you rate our product? (N1)  
 V9. How old are you? (N2)  
 V10. What do you feel would be a fair price for this product? (N6)  
 V11. What is your favorite brand? (A45)  
 V12. What could we do to make our product better? (A500)

If a respondent prefixes a numeric response with a dollar symbol (as might readily happen for V10), the dollar symbol will be eliminated from response when importing the Email file.

The basic text for an e-mail survey can be created by selecting Design, Email Survey. This will provide a foundation for the body of your email. It is expected that you will edit the text as necessary for your specific application and audience.

## Capturing a Respondent's Email Address

To capture the respondents' Email addresses, include a variable in the codebook that has the name EMAIL. Give it a format sufficient to hold all Email addresses (e.g., A70). The EMAIL variable may be placed anywhere in the codebook. Do not indicate anything on the Email survey itself. StatPac will properly capture the respondent's Email address when ever the EMAIL variable name is specified in the codebook.

Unlike plain-text Email, CGI Email cannot automatically capture the respondent's email address. Therefore, in order to capture the respondent's Email address you must explicitly specify it as a field on your web page.

## Filtering Email to a Mailbox

All e-mail programs allow you to automatically or manually send a received e-mail to a mailbox or text file. Automatic "filtering" is often used to sort the incoming Email into a series of mailboxes. These mailboxes are simply text files containing all the Emails that have been routed to the mailbox. The naming convention for the mailboxes is different for each Email program. Eudora and Outlook Express programs use a .mbx extension. The first step is to determine the name of the mailbox. That is, the name of the file that contains the completed surveys (i.e., that holds the filtered Emails). This is the file that you will import into StatPac. If you do not have an Email program that has filtering to mailboxes, you could import the incoming Email survey's into StatPac one at a time from the clipboard, but this might be slow. A better technique might be to use notepad to copy/paste all the Emails into one file that can then be imported into StatPac.

## General Considerations for Plain Text Email

Because of the wide diversity in e-mail readers, we suggest that you limit every line of your e-mail survey to 60 characters. This will avoid unintentional word wrap. Create your email using a non-proportional font (e.g., Courier) to make it perfectly clear as to you how long a 60 character line will be.

We suggest that HTML tags not be used to enhance the appearance of your e-mail. Many e-mail readers do not recognize the control sequences used by HTML tags, and these potential respondents will see control codes as part of their e-mail. Make your e-mail surveys generic looking so they will display properly with all e-mail readers.

Before actually sending out the surveys, send several copies of the e-mail survey to yourself. Make sure the surveys look exactly the way you want the respondent to see them. Then reply to each of the surveys. Now check your mail to make sure that they get filtered to the correct mailbox and check the import procedure to make sure that the data is being properly imported. Checking the entire import process is crucial for e-mail and Internet surveys.

# Internet Surveys

---

## Overview

StatPac can create single page or multiple page Internet surveys from a codebook. In order to use the Internet survey feature you must have a Web site that supports CGI. Usually, this means you will have access to a cgi-bin folder on your server. Nearly all hosting services support CGI, so you may need to contact your ISP for more information. StatPac has a Perl script that you will install in your cgi-bin folder. If you do not wish to host your own survey, StatPac can provide a server for you.

Optionally, you will have a WYSIWYG HTML editor. StatPac will create aesthetically pleasing and fully-functional Internet surveys, but you may want to visually enhance their appearance with graphics or other design features. In order to do that, you must have a **What-You-See-Is-What-You-Get** HTML editor. Microsoft Front Page and Macromedia Dreamweaver are examples of WYSIWYG HTML editors. However, any WYSIWYG HTML editor will work... even recent versions of Microsoft Word.

After StatPac creates the HTML pages, you may edit them, but if you were to subsequently regenerate the pages with StatPac, all edits would be lost. Therefore, don't edit the HTML until you are satisfied with StatPac's pages. Test the StatPac generated HTML pages online before using an HTML editor to enhance the appearance of the pages.

Many of the features of Internet surveys will only work when the survey is online. Things like branching, cookies, piping, popup windows, help windows, and page submissions will not work on your local computer. Use your local computer to view and edit the pages, but the survey must be on a server to test the functionality of these features.

---

## Internet Survey Process

The basic process for creating and using Internet surveys is as follows. We suggest that you follow this process for each Internet survey you conduct.

### Server Setup

If you will be using your own server to host the survey and you have not already setup your server, select **Server > Setup** to specify your server settings.

If you will be using StatPac's server to host the survey, you'll be able to select a private folder name when you design your first survey.

## Create the HTML Survey Pages

1. Create a codebook.
2. Create a default script by selecting Design > Internet Survey.
3. Modify the Primary Settings and script as necessary.
4. Generate and view the HTML files. Repeat this step as necessary.

## Upload the Files to the Web server

Select Server > Auto Transfer . Click the Upload Survey button.

## Test the survey

Test the survey by repeatedly taking it online as if you were a respondent. Test branching and validity checks.

## Download and import the test data

Select Server > Auto Transfer. Click the Download Responses button. The responses will be downloaded and imported into StatPac.

Alternatively, you may manually download the file of responses by selecting Server > FTP. In the top pane, navigate to the folder containing the responses (usually cgi-bin). Drag the response file from the top pane to the lower pane to download the file. After downloading the response file to your local computer, select Data > Import > Internet Response File to import the data into StatPac.

## Delete the test data from the server

Select Server > Auto Transfer Select the Delete tab and click the Delete Responses button.

Alternatively, you can manually delete the responses using FTP.

If you are using the StatPac server, select Server > FTP > StatPac. Select View > Response Folder. Right click on the response file and select Delete.

If you are using your own server, select Server > FTP > *YourServerName*. In the top pane, navigate to the folder containing the responses (usually cgi-bin). Right click on the response file and select Delete.

## Conduct the survey

Email invitations or somehow make respondents aware of the link to the survey.

## Download and import the data

If you are using your own server, select Server > Auto Transfer > YourServer. Click the Download Responses button.

If you are using StatPac's server, select Server > Auto Transfer > StatPac. Select the Auto Transfer tab and click the Download Responses button.



Alternatively, you may manually download the file of responses by selecting Server > FTP. In the top pane, navigate to the folder containing the responses (usually cgi-bin). Drag the response file from the top pane to the lower pane to download the file. After downloading the response file to your local computer, select Data > Import > Internet Response File to import the data into StatPac.

The server will always contain the entire data set unless you delete the response file on the server (i.e., downloading the response file does not erase it from the server). Therefore, you can download the data at any time from a survey in progress and the download would contain the entire data set from the beginning to that point in time. When you import the data, you would overwrite the existing file because the newly downloaded file contains the entire data set.

## Display a survey closed message

Select Server > Auto Transfer . Select the Delete tab and click the Delete Survey button. The survey will be deleted and the survey closed page will be shown to people attempting to access the survey. If you repeat this process, the survey closed page will be deleted.

Alternatively, you can use FTP to manually close a survey. Select Server > FTP. In the top pane, navigate to the folder containing the survey. Right click on the *SurveyName.htm* file and rename it to something else. Right click on the *SurveyName\_closed.htm* file and rename it to *SurveyName.htm*,

---

## Server Setup

Before you can create an Internet survey on your own server, you must tell StatPac about that server. Select Server > Setup.

The screenshot shows the 'Server Setup' dialog box with the following fields and options:

- Server Type:** Two radio buttons: ☒ Unix / Linux / Apache and ☐ NT / IIS / Windows.
- Connection:** Four text input fields: Domain Name, FTP Server, User Name, and Password.
- Paths and Folders:** Four text input fields: FTP Path to WWW Root Folder (public\_html), FTP Path to CGI Script Folder (public\_html/cgi-bin), Response Storage Folder (./), Server Path to Perl (/usr/bin/perl), Mail Method (a dropdown menu showing 'Unix Sendmail'), and Server Path to Mail Program (/usr/sbin/sendmail).
- Buttons:** At the bottom, there are navigation arrows, a 'New Server' button, a 'New' button (highlighted with a dashed border), and a 'Delete' button.

If you already have setup a server or servers, use the arrow keys to scroll through your server list.

To create a new server profile, click the New button. To delete the server profile that is currently displayed, click the Delete button.

Your ISP will be able to tell you the following FTP login and folder information.

## FTP Login Information

### ***Server Type***

There are basically two types of servers: 1) Unix / Linux and 2) Windows NT / IIS. When you make a Server Type selection, the most likely Paths and Folders settings will be filled in

### ***Domain Name***

The domain name should be specified without an http or www prefix. For example, *statpac.com* or *webpoll.org*.

### ***FTP Server***

The FTP Server is the address of the FTP server. It's usually your domain name with an ftp prefix. For example, *ftp.statpac.com* or *ftp.webpoll.org*. It might also be just the domain name (i.e., *statpac.com* or *webpoll.org*). It could even be an IP address.

### ***Username and Password***

Your Username and Password will be provided by your ISP. Usernames and passwords are usually case sensitive, so use care when entering the information.

## Paths & Folder Information

Web surveys will not function properly unless you get all of the settings right. There is a good chance that the default settings are correct, but not necessarily. So please be careful. On a Unix or Linux server, this information is case sensitive and is typically lower case.

### ***FTP Path to WWW Root Folder***

When you log in to your server using an FTP program, you'll be sitting in a folder on the server. Your wwwroot folder is the folder where you put your Web site HTML files. You should see your Web site home page in that folder. It might be the FTP login folder or it might be a subfolder. All of these are likely subfolder names.

*public\_html*  
*wwwroot*  
*docs*  
*yourdomain name.com*

If your FTP login folder is the same as your wwwroot folder, then leave this setting blank. If your wwwroot folder is in a subfolder, specify the subfolder name.

If you don't know, contact your ISP or try to discover it yourself by exploring your server. Select File > Exit. Select Server > FTP and select the server you are trying to set up. The top pane will be your FTP login folder. Do you see any HTML files in that folder? If not, do you see any of the above subfolder names? You can double

click on the subfolder names to see the contents of that folder. You're looking for the folder that contains files with extensions of .htm or .html (e.g., index.htm or default.html). Close the FTP program and return to the Server Setup program to enter the folder name.

### ***FTP Path to CGI Script Folder***

This folder is easy to identify because it is nearly always called cgi-bin or cgi. It is usually a subfolder of the wwwroot folder. Specify the path as the full path to the folder beginning at the FTP login folder.

#### **Example 1 – The wwwroot folder is a subfolder (typical Unix/Linux server)**

The server folder structure is:

FTP Login Folder  
public\_html  
cgi-bin

The FTP path to the wwwroot folder would be: public\_html

The FTP path to the CGI script folder would be: public\_html/cgi-bin

#### **Example 2 – The wwwroot folder is a subfolder (typical Windows IIS server)**

The server folder structure is:

FTP Login Folder  
wwwroot  
cgi-bin

The FTP path to the wwwroot folder would be: wwwroot

The FTP path to the CGI script folder would be: wwwroot/cgi-bin

#### **Example 3 – The wwwroot folder is the wwwrootfolder (e.g., godaddy.com)**

The server folder structure is:

FTP Login Folder is the wwwroot folder  
cgi

The FTP path to the wwwroot folder would be: leave blank

The FTP path to the CGI script folder would be: cgi

#### **Example 4 – The wwwroot folder is a subfolder called mydomain.com and the cgi script folder is at the same folder level as mydomain.com**

The server folder structure is:

FTP Login Folder  
mydomain.com  
cgi-bin

The FTP path to the wwwroot folder would be: mydomain.com

The FTP path to the CGI script folder would be: cgi-bin

## ***Response Storage Folder***

This is the folder where respondents' answers will be stored. It is specified differently depending of the type of server.

### **Unix or Linux Server**

On a Unix/Linux server, it is expressed as either an absolute server path or a path relative to the cgi script folder. **We highly recommend leaving the setting as ./ which means to store responses in the cgi script folder.** The cgi script folder is not visible to the outside world on a properly configured server.

There are numerous ways to specify the storage folder on a Unix or Linux server. The setting may be specified as absolute server path or relative to the cgi-bin folder. Absolute server paths always begin with a forward slash.

All of these would store the results in the cgi-bin folder.

`StorageFolder=/home/username/public_html/cgi-bin` (absolute server path)

`StorageFolder=../cgi-bin` (two periods)

`StorageFolder=./` (one period)

All of these would store the results in a folder called "storage" which is immediately below the cgi-bin folder.

`StorageFolder=/home/username/public_html/cgi-bin/storage`

`StorageFolder=../cgi-bin/storage` (two periods)

`StorageFolder=./storage` (one period)

These would store the results in a folder called "private" which is at the same level as public\_html (and therefore not accessible to the outside world).

`StorageFolder=/home/username/private` (absolute server path)

`StorageFolder=../../private` (two periods)

These would store the results file in a folder called "storage" which is immediately below the public\_html folder. Note that this may pose a security risk because the storage folder would be accessible to the world.

`StorageFolder=/home/username/public_html/storage` (absolute path)

`StorageFolder=/public_html/storage`

`StorageFolder=../storage` (two periods)

### **NT, Windows, or IIS Server**

On an NT or IIS server the setting is specified using a DOS path (i.e., the full path beginning with a drive letter to the folder where responses should be stored). Users must have read/write access to the folder. An example might be:

StorageFolder=c:\inetpub\wwwroot\cgi-bin

In order to use FTP to retrieve the data, the wwwroot folder name must part of the StorageFolder path. For example, you would not be able to use FTP to retrieve this data because the storage folder is not below the wwwroot folder.

StorageFolder=d:\datastorage

Some older NT servers require that you use double backslashes instead of single backslashes. If you receive a “Server Busy” message after clicking the submit button on a survey, try changing the path to double backslashes in place of single backslashes:

StorageFolder=c:\\inetpub\\wwwroot\\cgi-bin

### ***Server Path to Perl***

Type the absolute path where Perl is installed on your server. Your ISP will be able to tell you this information. The default settings are most likely correct. The syntax for this setting is different for Unix/Linux and NT/IIS servers.

#### **Unix or Linux server:**

Perl=/usr/bin/perl

#### **Windows (NT or IIS) server:**

Perl=c:\perl\bin\perl.exe

### ***Mail Method***

There are 4 mail methods to select from. The defaults are probably correct. For Unix/Linux servers, we suggest Unix SendMail. For NT/IIS we suggest SMTP Mail Server.

#### **Unix Sendmail**

Use this method on Unix/Linux servers only. Set the server path to the mail program to point to the absolute server path. Your ISP should be able to give you the path and name of your server mailing program. For example, usr/sbin/sendmail

#### **Perl Mail: Sendmail**

This method may be used with any kind of server. It uses the perl Mail::Sendmail module. You must have the perl module installed on your server to use this method..

### **SMTP Mail Server**

This method may be used with any kind of server. It uses your SMTP server to send emails.

### **Perl Net: SMTP**

This method may be used with any kind of server. It uses the perl Net:SMTP module. You must have the perl module installed on your server to use this method..

### ***SMTP Port***

When you choose one of the SMTP methods you must also specify the SMTP port. Port 25 is the default and it is most likely correct for your server although some servers use a different port. Your ISP to will be able to tell you your SMTP mail port number.

---

## **Design Considerations for Internet Surveys**

The first step in any survey is to create a codebook. Generally, this would be done using the Grid. There are only a few special considerations in designing a codebook for an Internet survey.

1. Use a short lower case codebook name without spaces, dashes, or special characters.
2. Keep your survey pages short. Responses are only collected when the user clicks the submit button. If the user gets frustrated and leaves your Web site without completing the survey, none of her responses will be recorded. You can dramatically increase response by keeping your surveys short (e.g., under 20 questions). If your survey needs to be longer, use a multiple page survey so that responses are stored at the end of each page. Even if a respondent fails to finish the entire survey, data will be captured for each page they completed.
3. Allow missing response for most items. When you do not allow missing response and the user clicks the submit button without answering all the items, they will be presented with a message to complete the missing item. If they become frustrated and leave your Web site, none of their responses on that page will be recorded even though only one item might actually be missing.
4. Specify a variable name for each variable. Do not use special characters in the variable name except the underscore character. A good variable naming scheme is Q1, Q2, Q3a, Q3b, etc.
5. When creating multiple response variables, there must be the same number of variables as value labels, and all the value labels must be specified for each of the variables of the multiple response variables. For example, if you have a survey with a question that says "Check all that apply", and there are five response choices (value

labels), then the codebook must contain five variables with identical variable and value labels.

6. Limit branching to variables that will use radio buttons. Branching out of text boxes or check boxes is not supported. Both simple and complex branching are supported.

7. Test your survey online before going live. This involves completing the survey several times and importing the data into StatPac. Do not assume that if the survey visually looks okay, it is okay. When you test the survey online, specify an answer for every question. For multiple response checkboxes, check every box. This is the only way to guarantee that you have not made any errors. Visually inspect the .asc response file (using notepad). If any numeric fields have more than one value (with a comma separator), it means that you have made an error in the codebook or StatPac script (two variables have the same name or the variable is specified twice in the script).

8. Test your email invitation by sending an invitation to yourself. Make sure the link in the invitation works like you expect it to. In other words, try it!

Testing is a mandatory component of every internet survey. Do not bypass this step!

---

## Special Variables for Internet Surveys

There are three special variables for Internet surveys: *IPAddress*, *Today*, and *RespondentID*. If you add these variables to your codebook, you will be able to capture the IP address of the respondent, the date that they completed the survey, and a unique Respondent ID number. After generating an Internet survey, StatPac will ask if you want to add these variables to the codebook. In most cases, you should answer yes.

Alternatively, you can manually add these variables to the codebook during the study design. The *IPAddress* and *RespondentID* variables should have an A15 format and the *Today* variable should have an N8 format. When capturing the date, it will be stored in the data file in YYYYMMDD format. These variables may be placed anywhere in the codebook. They will not be shown on the web pages and are for internal use only.

There is an additional N5 variable, *Seconds*, that will be automatically added to Internet survey codebooks. This variable is used to hold the number of seconds it took the respondent to complete the survey. If a respondent did not complete the survey, the *Seconds* variable will be blank. The name for the *Seconds* variable is set in the StatPac.ini file with the SecondsVarName = parameter. If the parameter is left blank, then seconds will not be added to the codebook or calculated.

The *RespondentID* variable can be used to match respondent's data with an existing data base of information. First, include *RespondentID* as an A15 variable in the codebook. When you generate the Internet survey, it will not appear on the survey.

The *RespondentID* variable must be included in the codebook if you intend to track who responded to the survey. In a typical web survey, you would use StatPac's bulk e-mail program to send potential respondents an invitation to take your survey and there would be a link in the e-mail to the survey URL. In order to track who responded or to match respondents' data with the data base, the URL link must be appended with a question mark and the respondent's ID number. For example, when sending the e-mail to respondent whose ID in the data base was 91246, you would use this as the link URL in the email to that respondent. The respondent ID may consist of any alpha or numeric characters.

<http://www.yourdomain.com/surveyname.htm?id=91246>

Note: If you are using password protection for the survey, the link might be:

<http://www.yourdomain.com/cgi-bin/surveyname.pl?id=91246>

StatPac's bulk e-mail program will automatically append an ?id= to the URL link in the e-mail invitation. If you use StatPac to send email invitations, ID handling is automatic.

---

## Script to Create the HTML

The second step in creating an Internet survey is to create a script that defines all the characteristics of the HTML pages. The script is a set of commands that tells StatPac how to generate the HTML survey files that will become your Internet survey. The script language is quite easy to understand, and there are only a few commands that you'll need to know. In most cases, the default script created by StatPac will require only minor editing.

To create a default script, first open the codebook. Select Design > Internet Survey and the Internet script window will show the current script. If a script has not been previously created for this codebook, a default script will be created. The default script is StatPac's best guess of how you want your survey to look, but in most cases you'll be able to improve on its appearance by editing the script.

Again, you do not need to know or understand all the script commands. Usually minor editing will be sufficient. The script is divided into sections to make it easier to understand. There are three major sections: Primary Settings, Advanced Settings, and Survey Creation.

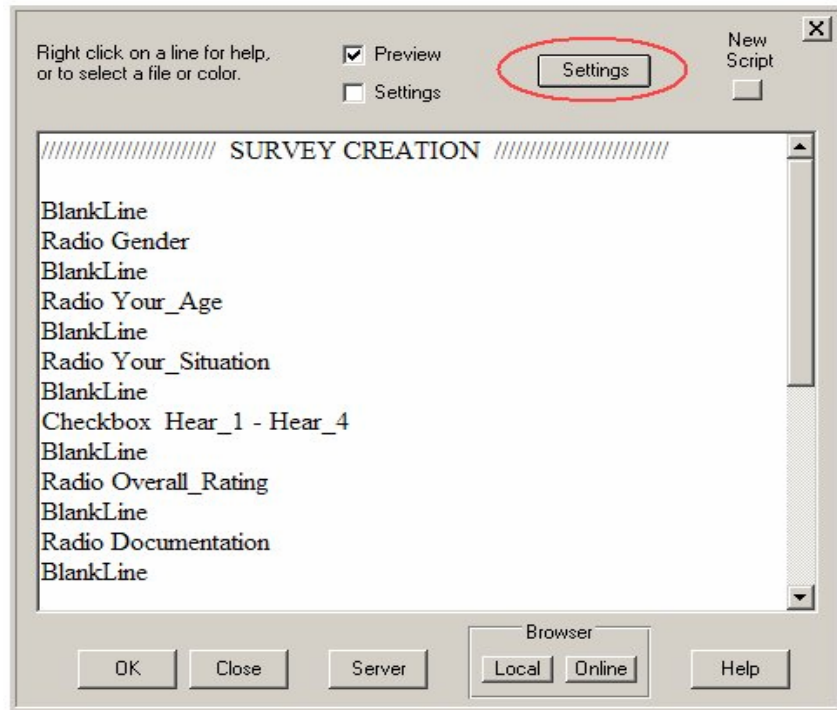
The Primary settings must be specified for each survey. They control parameters that are unique to a given survey.

The Advanced settings control text attributes such as fonts, colors, and spacing. These settings often remain the same from one survey to another. The Advanced settings can be saved in a style sheet so they can be used in a future survey.

The Survey Creation section has commands to control the order and appearance of objects (i.e., radio buttons, check boxes, text boxes, etc.).

When Settings is not checked, only the Survey Creation section will show in the script. When Settings is checked, all the settings will be shown. The Primary and Advanced settings may be edited directly in the script window (if Settings is checked), or you may click the Edit button to use the Script editor.

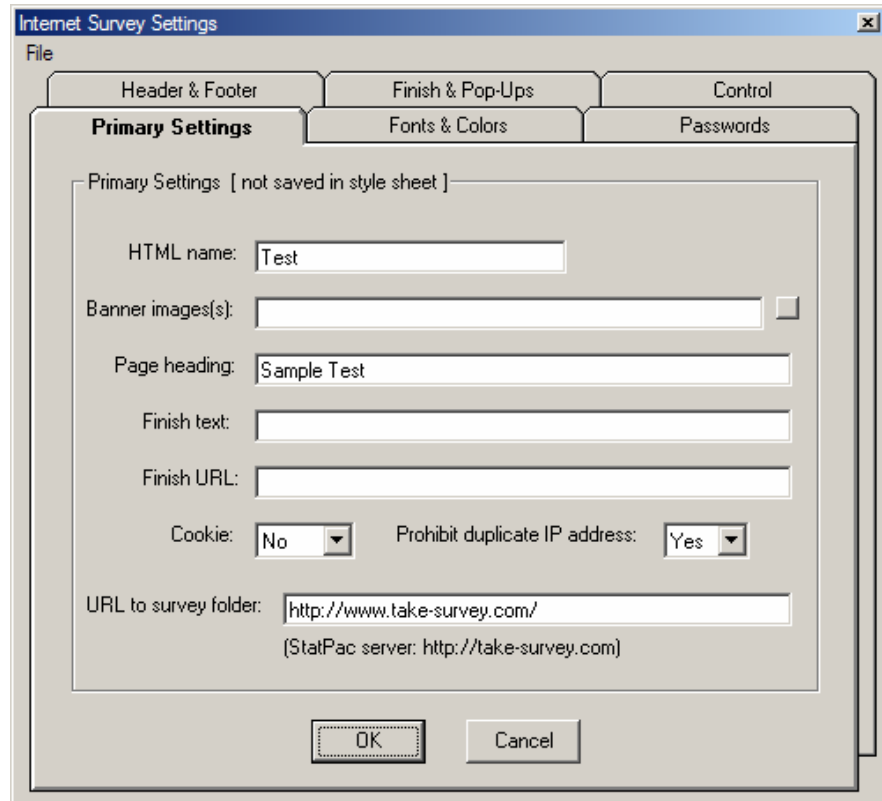




The basic process is to let StatPac create a default script and generate the Internet survey HTML pages. Then view the survey pages and note the things you would like to change. Make changes in the script and regenerate the pages. View the HTML pages again and continue making changes to the script until you are satisfied with the survey.

Once you are satisfied with the appearance of a survey, you can click the Server button to upload it to the Internet.

You will want to make changes to the Primary Settings, but only the URL to survey folder is critical because it defines the server and folder that will host the survey online.



## Command Syntax & Help

Most script commands have two parts. The part to the left of the equals symbol is a keyword for the script. The part to the right of the equals symbol is the text for the keyword. If the text part (to the right of the equals symbol) is longer than one line just continue typing without pressing [Enter], so the text automatically wraps to the next line(s). Unlike a procedure file, an indented line will not be interpreted as a continuation of the previous line.

While viewing the script, you can right click on any line to learn more about that command. If the command involves a file or color selection, the right click will also offer a settings choice.

The actual window for the script has standard editing features. Use Ctrl X, Ctrl C, and Ctrl V to cut, copy, and paste text within the window. There is also a semi-automatic copy and paste feature to expedite changes to the script. You may highlight text from other selected areas of the screen and then click in the script window to automatically paste them into the text (without actually selecting copy and paste). The copy will work from the workspace or the Detail window. To copy a variable name, select the variable on the right from the list and double-click in the script window on the line where the variable name should appear. Variable names will always be added to the end of the line.

Once created for a given codebook, the script is automatically saved. If you subsequently select Design, Internet Survey for the same codebook, the previously created/edited script will be shown. The script itself is an ASCII text file with the name "*codebookname.script*". The script is created from the codebook, but once created, it is independent of the codebook. For example, say you are working on a

codebook and then create a script just to see what the Internet survey will look like (so far). Then you go back to the Grid and add more variables. The next time you select Design, Internet Survey, the previous script will be shown -- without the new variables. Click the New Script button to recreate Survey Creation portion of the script with the new variables. Alternatively, you can manually add the commands to the Survey Creation portion of the script.

If you want to completely start over with the default script for a given codebook, close StatPac, delete the *codebookname.script* file, rerun StatPac, load the codebook, and select Design, Internet Survey.

---

## Saving and Loading Styles

The Advanced settings control the "look and feel" of a survey. Fonts and colors are part of the advanced settings.

You can save the "style" to a file so that you can recall and use the style on a future survey. The style includes most of the advanced settings (colors, and page layout parameters). The "Style Buttons" let you save the current style or load a previous style. "Style files" have the extension of .style and the default folder for style files is the installation folder (although you can save or load styles to and from any folder).

While working on a survey, save the style by clicking on the Style Save Button and typing a name for the style. Load a previous style by clicking on the Style Open Button and select the style. When you generate the HTML files, the current style will control the appearance of the survey.

The actual style file is an ASCII text file that contains most of the Advanced settings. When you first select Design > Internet Survey, StatPac loads a style called Default.style from the installation folder. If you want to change your default style, save the desired style to the installation folder using the name "Default" and overwrite the existing Default.style.

---

## Survey Generation Procedure

Generally, the procedure you'll follow will be to first click the OK button. This will run the script that creates the Internet survey. StatPac will create several HTML pages: a loader page, one for each page of the survey, a thank-you page, and a survey-closed page. Other HTML files, including help and popup windows, and a cookie-cutter might also be created.

When the Preview box is checked, the survey will be shown in a preview window. In the Preview window, select View to select the page you want to look at.

You may also select the Browser Local button to launch Windows Explorer and view the survey you created. If your survey contains multiple pages, you will have to look at each page individually. After examining the appearance of the survey, close Explorer. If necessary, make changes to the script and repeat the process. You can continue making changes to the script until you are satisfied with the appearance of the Internet survey. It is important to note that many features (including the continue and submit buttons) will not work properly until the files are actually uploaded to the Internet.

---

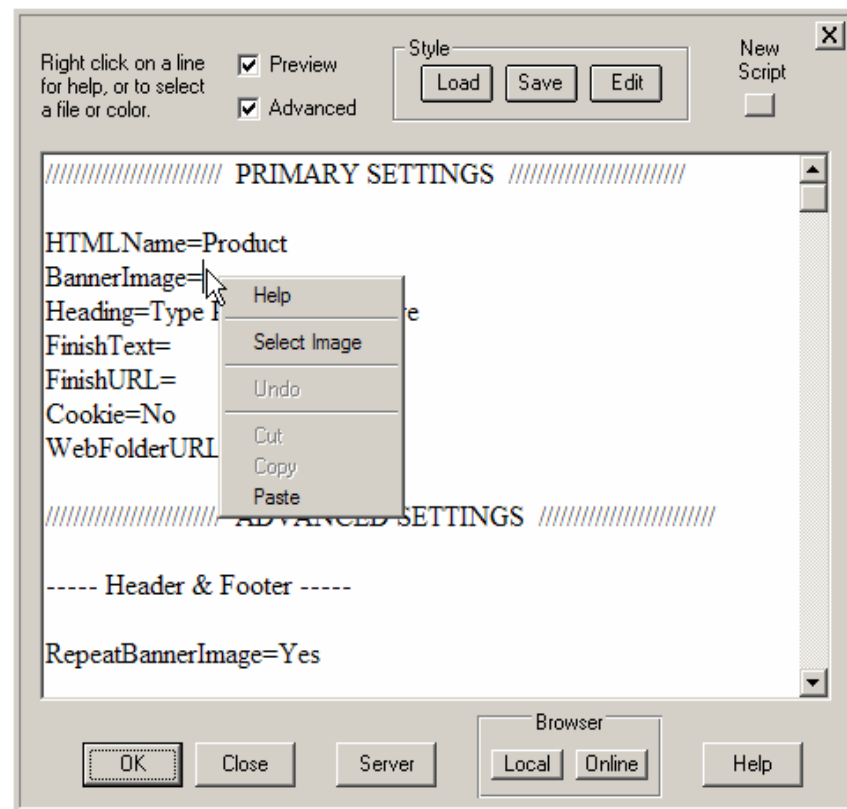
## Script Editor

Both the Primary Settings section and the Advanced Settings section of the script can be edited using the Script editor. For inexperienced users, this will be easier than editing the script directly, although both methods achieve the same goal.

The Script Editor offers the ability to make changes to the Primary and Advanced sections of the script using a form interface. The tabs represent different sections of the script. Changes made using the Script Editor will be reflected in the script itself when you exit the script editor.

In the Primary and Advanced settings section of this manual that follow, the keyword in the script is specified. When using the Script editor, you do not need to be concerned with the keywords themselves.

Click the Edit button to evoke the Script editor. Right click on any line in the script editor to get help for that command.



---

## Imbedded HTML Tags

HTML tags may be imbedded in text settings to control the appearance of the text. These are:

Start and stop bold: `<b>` and `</b>`

Start and stop underlining: `<u>` and `</u>`

Start and stop italics: `<i>` and `</i>`

Insert a line break: `<br>`

Thanks=`<b>`This entire text is bold.`</b>`

Text=Only one word is `<u>`underlined`</u>`.

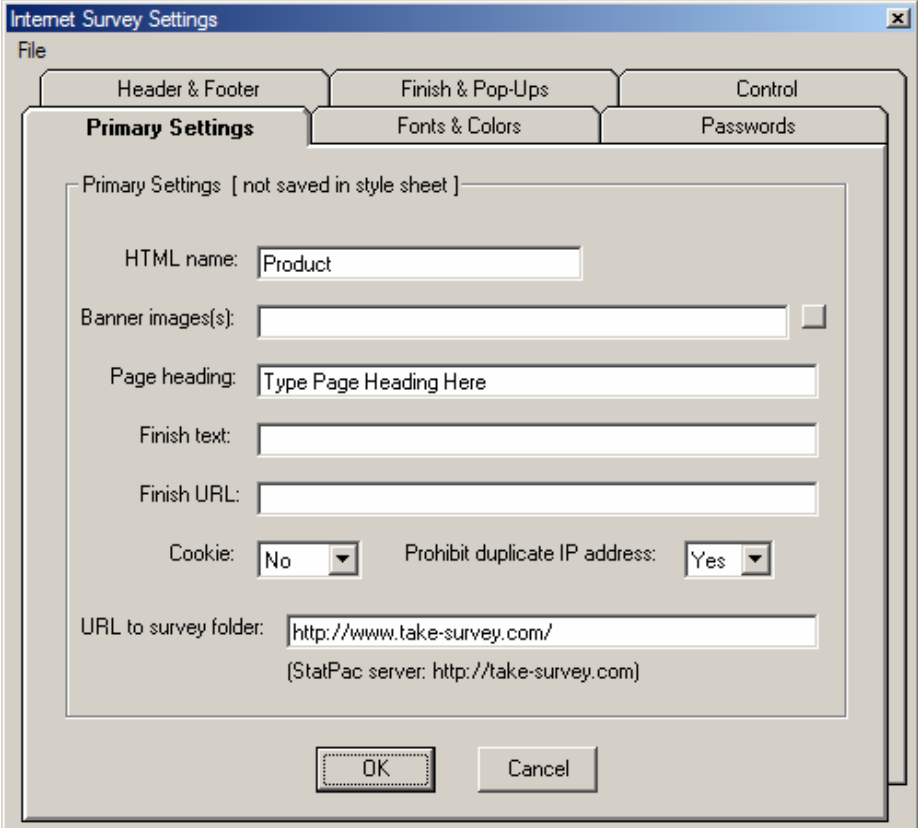
Instructions=`<b><i>`This is bold & italics.`</i></b>`

Closed=Thank you for your interest.`<br>` `<br>` The survey is closed.

---

## Primary Settings

The Primary settings will always be shown at the beginning of the script. This is the only section of the script that you must complete. It specifies critical information that is likely to vary from survey to survey. There are eight Primary settings.



The screenshot shows the 'Internet Survey Settings' dialog box with the 'Primary Settings' tab selected. The dialog has a menu bar with 'File' and several tabs: 'Header & Footer', 'Finish & Pop-Ups', 'Control', 'Primary Settings' (active), 'Fonts & Colors', and 'Passwords'. The 'Primary Settings' section contains the following fields and controls:

- HTML name:
- Banner images(s):  ☐
- Page heading:
- Finish text:
- Finish URL:
- Cookie:  (dropdown arrow)
- Prohibit duplicate IP address:  (dropdown arrow)
- URL to survey folder:   
(StatPac server: <http://take-survey.com>)

At the bottom are 'OK' and 'Cancel' buttons.

### HTML Name

HTML Name sets the name for all survey pages. The default will be the same as the codebook name but you may change it. This will be the name of the survey on the Web and it will be part of the link to the survey. An example would be:

`HTMLName=Research`

The first viewable page of the survey is named *HTMLName\_1.htm*. All subsequent pages of the survey (including the thank-you page) will have file names with an underscore and number suffix. The last numbered file is the thank-you page, which is the page that respondents will be shown when they click the final submit button.

A one-page survey would have the following files.

Research.htm (Loader)  
Research\_1.htm (1st page)  
Research\_2.htm (Thank-you page)

A three-page survey would have the following files.

Research.htm (Loader)  
Research\_1.htm (1st page)  
Research\_2.htm (2nd page)  
Research\_3.htm (3rd page)  
Research\_4.htm (Thank-you page)

Additionally, the HTMLName command is used to name several other files.

Research\_closed.htm (Survey is closed page)  
Research\_cookie\_cutter.htm (Delete the cookie)  
Research\_popup\_1.htm (1st popup window)  
Research\_popup\_2.htm (2nd popup window)  
Research\_help\_1.htm (1st help window)  
Research\_help\_2.htm (2nd help window)  
Research\_start.htm (Loader page for password protected surveys)

The HTMLName\_closed.htm page can be used after a survey has been closed. After a survey is closed you can delete the survey from your server. However, you probably also want to prevent late responders from getting a page not found message. Therefore, when you delete a survey, the survey closed page will be shown to respondents.

When cookies have been used to prevent visitors from taking the survey more than once, then you also need to upload a file named codebookname\_cookie\_cutter.htm. This file is necessary to test your installation. When a respondent finishes a page of the survey, they will be given a cookie as they advance to the next page. The cookie contains the ID number, and controls whether they will be able to return to a previous page and it will redirect their browser to the proper page if they quit the survey without completing it and come back to finish it at a future time. When you test your survey online, you too will receive the cookie. Thus, you could test it once but you might be unable to test it again. To delete the cookie from your computer, set your browser to *HTMLName\_cookie\_cutter.htm* and the cookie will be deleted. You'll then be able to test the survey again.

## Banner Image

BannerImage sets the image that will be shown at the top of the page. To select the image, right click on the BannerImage command line. If you're doing a survey for a client, BannerImage is probably your client's logo.

**BannerImage=c:\images\logo.jpg**

Multiple banner images may be show side by side. After right clicking on the line, select the first banner image. Then right click on the line again and select the second banner image. They will be separated on the command line by a semicolon.

**BannerImage=c:\images\logoA.jpg;c:\images\logoB.gif**

To erase any or all previously selected banner images, simple delete them from the BannerImage command line.

Tip: To capture a client's logo, go to their Web site. Right click on their logo and select Save Picture As. You may need to edit the logo with image editing software such as Photo Shop or Paint Shop Pro.

## Heading

Heading sets the text for the page heading. It is generally the title of the survey.

**Heading=Acme Inc. Employee Survey**

## Finish Text & Finish URL

FinishText sets the text for a link on the thank you page and FinishURL sets the URL for the link. If you are doing a survey for a client, this is probably a link to their home page. If the survey is for your company, it's probably a link to your home page.

**FinishText=Click here for the StatPac home page**  
**FinishURL=http://statpac.com**

If both are left blank, then the finish page will not have an outgoing link. If the FinishText is specified and the FinishURL is left blank, the text on the thank-you page will appear as text only without a hyperlink.

## Cookie

Cookie sets the type of cookie that will be used to prevent multiple submissions from the same computer. The valid settings are: None, ID, Partial, or Full.

**Cookie=None**

**Cookie=ID**

The respondent's computer will be given a cookie so that a respondent who attempts to take the survey multiple times will be assigned the same ID number as previous administrations. If they finish the survey or quit prematurely and attempt to take it again, they will start at the first page of the survey and will be assigned the same ID number as their first access. If they change previously entered data, their most recent entry will be captured by the program.

### ***Cookie=Partial***

A cookie will be given when the respondent reaches the thank-you page to prevent them from taking the survey again. Thus, the respondent will be able to page back and forth within the survey, but not after they have completed the survey. Once they've completed the survey, they will not be able to access it again.

### ***Cookie=Full***

A cookie will be given when the respondent submits each page. Thus, if a respondent stops taking the survey on a given page and tries to take the survey again at a future time, their browser will automatically be redirected to the page where they left off. Setting *Cookie=Full* turns the survey into a "forward only" survey. Respondents will be able to go forward to the next page of the survey, but they will not be able to go back to a previous page. In other words, their browser's Back Button will not work. Once they've completed the survey, they will not be able to access it again.

## **IP Address Control**

IPControl sets whether IP addresses will be used to prevent multiple submissions from the same IP address. The valid settings are: Yes or No

### ***IPControl=Yes***

When a respondent completes the survey, their IP address will be recorded in an IP log file. If they attempt to take the survey again, their IP address will be found in the log file and they will be prevented from taking the survey again.

IPControl is more effective than cookies because cookie blockers are sometimes installed on respondents' computers. IPControl cannot be blocked. However, IPControl should only be used when you have reason to believe that each potential respondent has a different IP address. For example, if you are surveying different companies and you want to allow only one respondent from each company, IPControl could be used. IPControl would not be appropriate if you were surveying multiple people from the same company.

When using cookie or IP control, the *HTMLName\_cutter.htm* will allow you to delete the cookie or your IP address. Since the *HTMLName\_cutter.htm* can be accessed from your browser, it means that anyone can access it. The *HTML\_SecureCutter* setting in the StatPac.ini system defaults file can be used to control whether you will be required to enter your StatPac serial number to access the *HTMLName\_cutter.htm* file. If *HTML\_SecureCutter=1*, you will be required to enter your serial number and if *HTML\_SecureCutter=0*, you will not.

## **Allow Cross Site Access**

Cross site access might be a useful feature. You can place the loader page on your domain (the URL respondents will see) and host the survey itself on the StatPac server. The loader page is named *HTMLname.htm* and its purpose is to load the first



page of the survey (named *HTMLname\_1.htm*). When cross site access is allowed, the loader page can be hosted on a server that is different than the survey itself.

Allowing cross site access has risks because it makes your survey more convenient to hacking attempts. StatPac was written to minimize the possibility of being hacked, and we are not aware of any successful attempts. However, we are aware of many situations where hackers have tried to defeat StatPac's security. This is especially true for surveys from financial institutions and "popularity polls" for the young adult audience. Allowing cross site access makes it more convenient for someone to try to hack the survey, so we recommend not using cross site access unless you actually need it.

The *HTML\_AllowCrossSite* setting in the StatPac.ini system defaults file is used to control cross site access. When *HTML\_AllowCrossSite* = 1, cross site access will be allowed. When *HTML\_AllowCrossSite* = 0, an *Access Denied* message will be displayed,

## URL to Survey Folder

WebFolderURL sets the server and optionally the folder where the survey will reside. It is the full URL to the folder that will hold the survey. All of the survey pages are uploaded to this folder.

### ***If you will be using the StatPac server:***

Specify StatPac as the WebFolderURL setting:

**WebFolderURL=take-survey.com**

If you want to use StatPac's secure SSL server, add an *https://www.* prefix.

**WebFolderURL=https://www.take-survey.com**

When you click OK to generate the HTML, the setting will be changed to reflect your current private folder on the StatPac server, and the HTML will be created using the modified setting. When using SSL, the *www.* prefix is required and will be added by the software if you inadvertently omit it when using *https://*.

The link to your surveys on the StatPac server will be:

**http://take-survey.com/foldername/surveyname.htm**

or

**https://www.take-survey.com/foldername/surveyname.htm**

To change your private folder name (when Settings is checked), right click on the WebFolderURL line and select Server Folder Setting. Alternatively, select Server>Auto Transfer and click the Folder tab.

After you change your folder name, you must regenerate the HTML so that the survey incorporates the new folder name and not the old folder name. The WebFolderURL

setting will be adjusted when to your new folder name when you generate the HTML.

***If you will be using your own server:***

Specify the full URL to the folder that will hold your surveys.

If your domain name is acme.com and you place the survey in the home directory, then you would set this parameter to:

**WebFolderURL=http://www.acme.com**

If you want to place your survey in a "survey" folder immediately below the home directory, then you would set WebFolderURL to:

**WebFolderURL=http://www.acme.com/survey**

If you want to run the submission process over a secure (SSL) server, then you must use the fully qualified secure socket URL:

**WebFolderURL=https:// www.acme.com /survey**

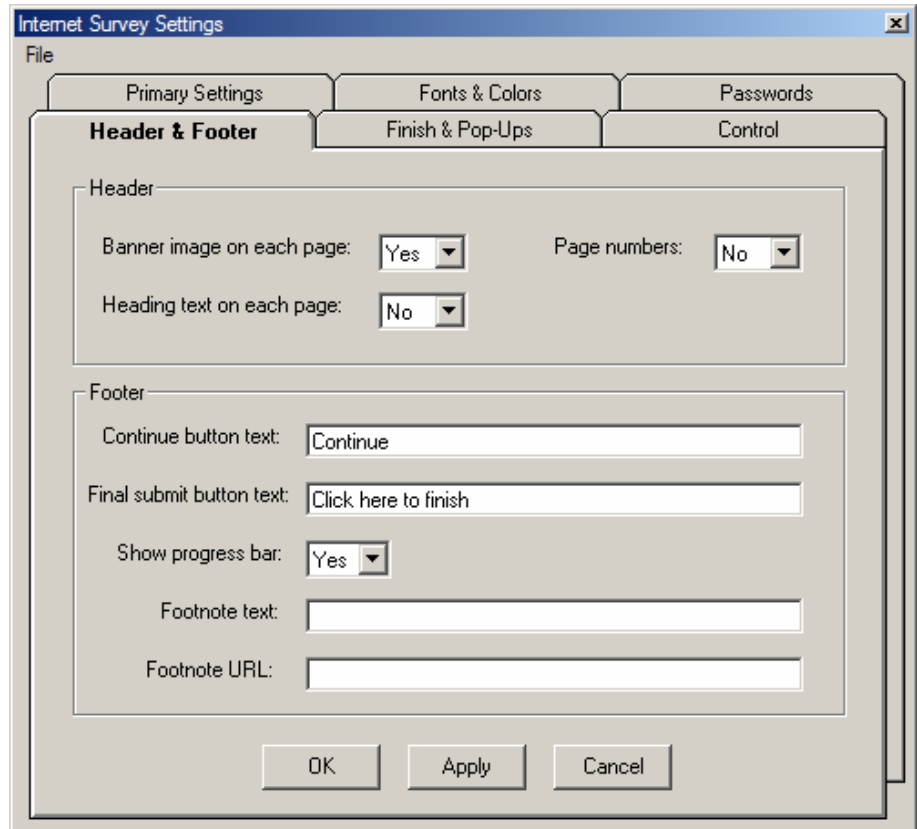
---

## Advanced Settings

The Advanced Settings can be viewed in the script by checking the Advanced Settings checkbox. However, it is usually easier to click the Settings Button.

### Header & Footer

The Header & Footer settings let you control what will appear at the top and bottom of each page.



### ***RepeatBannerImage***

RepeatBannerImage sets whether the banner image (as defined in the Primary Settings) will be repeated on each page. RepeatBannerImage may be set to Yes or No.

RepeatBannerImage=Yes

### ***RepeatHeading***

RepeatHeading sets whether the page heading (defined in the Primary Settings) will be repeated on each page. RepeatHeading may be set to Yes or No.

RepeatHeading=No

### ***PageNumbers***

PageNumbers sets whether page numbers will be shown at the top of each page. When a banner image is displayed, page numbers will appear in a small font below the banner image. If no banner image is displayed, they will appear below the page heading. PageNumbers may be set to Yes or No.

PageNumbers=No

### ***ContinueButtonText***

ContinueButtonText sets the text on the continue button for multiple page surveys. On a single page survey, this setting is ignored.

ContinueButtonText=Continue

The ContinueButtonText may be used more than once in the Survey Creation section of the script to change the continue button text on each page.

<Commands to create the first page go here>

ContinueButtonText=Click here for the second page

NewPage

<Commands to create the second page go here>

ContinueButtonText=Click here for the third page

NewPage

### ***SubmitButtonText***

SubmitButtonText sets the text on the final submit button for the last survey page. Clicking this button will take the respondent to the thank you page.

SubmitButtonText=Finish

### ***ProgressBar***

ProgressBar sets whether a progress bar will be shown at the bottom of each page. It applies only to multiple page surveys. The progress bar uses two graphics, blue.gif and grey.gif. Thus, when using the progress bar, both of these graphics must be uploaded to the same server folder as the survey. Auto Transfer will automatically upload these files when necessary. ProgressBar may be set to Yes or No.

ProgressBar=Yes

### ***FootnoteText & FootnoteURL***

FootnoteText sets the text for a link that will be shown at the bottom of each page and FootnoteURL sets the URL for the link. This is usually a link that a respondent can click if they have problems with the survey. If both settings are blank, no footnote will be shown. If FootnoteText is specified and FootnoteURL is left blank, the footnote will appear as text only.

FootnoteURL=http://statpac.com

FootnoteURL=mailto:admin@statpac.com?subject=Survey Help

## Finish & Popups

The Finish and Popup settings let you control the message respondents will see when the survey is completed, and the characteristics of popup windows if they are used.

The screenshot shows the 'Internet Survey Settings' dialog box with the 'Finish & Pop-Ups' tab selected. The 'Completion of survey' section contains two text input fields: 'Thank you message:' with the text 'Thank you. <br> Your response has been received.' and 'Survey is closed message:' with the text 'Thank you for your interest. <br> The survey is close'. The 'Pop-up & help windows' section contains four controls: 'Window height:' with a value of 250, 'Window width:' with a value of 350, 'Help link text:' with the text 'Help', and 'Default link text:' with the text 'Click Here'. There are also two dropdown menus: 'Show banner image(s) in pop-ups:' set to 'Yes' and 'Maximize pop-up windows:' set to 'Yes'. At the bottom are 'OK', 'Apply', and 'Cancel' buttons.

### Thanks

Thanks sets the text for the thank you page. The text will use the Title command attributes but may be modified using basic html tags to control the text appearance and line spacing.

Thanks=Your response has been received.<br>Thank you for completing the survey.

### Closed

Closed sets the text for the survey closed page. By default, the text will use the Title attributes, but may be modified using basic html tags to control the text appearance and line spacing.

Closed=Thank you for your interest.<br> The survey is closed.

### ***HelpWindowWidth & HelpWindowHeight***

HelpWindowWidth sets the Help window width in pixels and HelpWindowHeight sets the Help window height in pixels.

HelpWindowHeight=250

HelpWindowWidth=350

### ***HelpLinkText***

HelpLinkText sets the text that the respondent will see on the survey for the help link. If not specified, the default is "Help".

HelpLinkText=Click here for help

Help commands may be used together to insert a pre-existing help file. The keyword "HelpWindow" determines where the link will be inserted.

HelpWindowHeight=600

HelpWindowWidth=1000

HelpLinkText=Click here for help.

HelpFileName=help-file-name.htm

HelpWindow

### ***LinkText***

LinkText sets the text for the Link command which is used to insert a hyperlink into the survey. LinkText and LinkURL should be set before using the Link command.

LinkText=Click here to see our home page

LinkURL=http://www.statpac.com

Link

### ***PopupBannerImage***

PopupBannerImage sets whether the banner image will be displayed at the top of popup windows. PopupBannerImage may be set to Yes or No.

PopupBannerImage=Yes

### ***PopupFullScreen***

PopupFullScreen sets the size of popup pages. PopupFullScreen may be set to Yes or No. When set to Yes, popup pages will fill the respondent's entire screen. When set to No, popup pages will only fill the current window display area (i.e., the browser toolbar and URL address bar will not be obscured by the popup page). When there are no popup pages, the setting is ignored. A full screen popup is preferred if the popup window contains more than a couple of questions.

PopupFullScreen=No

## Control

The Control tab lets you change settings that control the basic operation of the survey

The screenshot shows the 'Internet Survey Settings' dialog box with the 'Control' tab selected. The 'Operation & control' section contains the following settings:

Setting	Value
Data capture method:	File
Maximize survey window:	Yes
Break out of HTML frame:	No
Hide ID number:	Yes
Allow survey to be cached:	No
Automatic variable advance:	No
Search engines indexing:	No
Delay branching:	No
Your email address:	you@yourdomain.com
Restart Seconds:	0
Extra spacing:	No
Radio text position:	Left
Area box text position:	Left
Area box position:	Center
Area box progress bar:	No

Buttons at the bottom: OK, Apply, Cancel.

### Method

Method sets the method that will be used to capture respondents' answers. Method may be set to Email, File, or Both. When Method=Email, the responses will be emailed to you. When Method=File, they will be saved in a file on the server. When Method=Both, responses will be emailed to you and stored in a file on the server. The suggested method is File.

**Method=File**

### Email

Email sets the suffix or full email address where the completed survey responses will be mailed. This command is ignored when using the File method.

In the first form of the command, only the suffix is specified and the email will be sent to your codebook name at the specified domain. For example, if you create an Internet survey for a codebook named "research", the completed responses will be mailed to research@domain.com. In the second form of the command, the entire email address is specified.

EMail=@acme.com  
EMail=John.Doe@acme.com

### ***RestartSeconds***

RestartSeconds sets the number of seconds that the thank you page is displayed before the survey is restarted. This feature is useful for kiosk surveys or when using a Web survey as the mechanism for data entry. When RestartSeconds is set to blank or zero, the thank you page will be displayed indefinitely. When set to a value greater than 0, the thank you page will show for the specified number of seconds, and then the survey will be loaded again beginning with the first page.

RestartSeconds=15

### ***MaximizeWindow***

MaximizeWindow sets whether the survey will attempt to maximize the respondents browser window to full screen. MaximizeWindow may be set to Yes or No.

MaximizeWindow=Yes

### ***BreakFrame***

BreakFrame sets whether the survey will attempt to break out of an HTML frame. Do not use this feature unless you are linking to the survey from within a frame set. BreakFrame may be set to Yes or No.

BreakFrame=No

### ***AutoAdvance***

AutoAdvance sets whether the screen will automatically scroll to the next question when a radio button is clicked. It is often disconcerting for respondents to see the screen scroll on its own, so the recommended setting is No. AutoAdvance may be set to Yes or No.

AutoAdvance=No

### ***BranchDelay***

BranchDelay sets whether branching is immediate or delayed until the page is submitted. BranchDelay may be set to Yes or No. When the last variable on a page contains a branch, it will always be delayed until the page is submitted.

When set to No, all other branching will happen immediately when the radio button is clicked. When set to Yes, all branching on that page will be delayed until the respondent finishes the page and clicks the submit button. BranchDelay may be used multiple times to change the setting from page to page or within a page.

BranchDelay=Yes



## **Cache**

Cache sets whether the respondents' browsers will cache the survey pages. Cache may be set to Yes or No. If you expect respondents to frequently use the back button to review previous answers, set Cache to Yes so their pages load faster. Otherwise, the setting is not important.

Cache=Yes

## **Index**

Index sets whether search engine spiders will be allowed to index and follow links on the survey pages. Index may be set to Yes or No. Each survey page will include the robots meta tag with instructions to search engine spiders to include or exclude the page from their index. Some search engine spiders ignore meta tags so setting Index to No will not guarantee that a page will not be indexed.

Index=No

## **ForceLoaderSubmit**

ForceLoaderSubmit sets the method used to load the first page of the survey. ForceLoaderSubmit may be set to Yes or No. The link to the survey is actually a link to a loader page. The loader can display the first page of the survey using two different methods.

When set to No (fastest method), the loader page is replaced with the first page of the survey. Using this method, you will not be able to tell how many people just looked at the survey without completing any questions. Data will only be captured when they click a submit button.

When set to Yes, the loader page is processed as if a submit button were clicked. The respondent's IP address and data and time are captured. Even if they don't complete any actual survey questions, you'll know that they looked at first page of the survey.

ForceLoaderSubmit=Yes

## **ExtraTallBlankLine**

ExtraTallBlankLine sets the height when the BlankLine command is used. ExtraTallBlankLine may be set to Yes or No. A setting of Yes increases the height of a blank line, and No decreases it.

ExtraTallBlankLine=No

## **RadioTextPosition**

RadioTextPosition sets the position of the text that is adjacent to a radio button. RadioTextPosition may be set to Left or Right. When set to Left, the text will appear to the left of a radio button, and when set to Right, the text will be displayed to the right of the radio button.

Example: RadioTextPosition=Right

RadioTextPosition may be used multiple times to display some questions with the text to the left of the radio buttons and some questions to the right of the radio buttons. For example, these lines would show two groups of questions with radio buttons in a horizontal format. The first group would have the text (questions) appear to the right of the radio buttons, and the second group would have the text appear to the left of the radio buttons.

```
Question=Please indicate the importance of these items:  
RadioTextPosition=Right  
Radio Impt_1 - Impt_5  
BlankLine  
RadioTextPosition=Left  
Question=Please indicate your satisfaction with the following items:  
Radio Sat_1 - Sat_5
```

### ***TextBoxTextPosition***

TextBoxTextPosition sets the position of the text that is adjacent to a text box. TextBoxTextPosition may be set to Left or Right. When set to Left, the text will be to the left of the text box, and when set to Right, the text will be displayed to the right of the text box. TextBoxTextPosition may be used multiple times to display some questions with the text to the left and other questions with the text to the right of the text box.

```
TextBoxTextPosition=Left
```

### ***LargeTextBoxPosition***

LargeTextBoxPosition sets the position of a large text box. LargeTextBoxPosition may be set to Left or Center. When set to Left, a text box will be left justified, and when set to Center, a large text box will be centered on the page. LargeTextBoxPosition may be used multiple times to show some large text boxes to the left and some centered. Applies to large text boxes only (i.e., multiple lines).

```
LargeTextBoxPosition=Left
```

### ***LargeTextBoxProgressBar***

LargeTextBoxProgressBar sets whether a large text box will have a progress bar underneath of it to show respondents how much of the maximum text space they have typed. LargeTextBoxProgressBar may be set to Yes or No. If the field width of the variable is very long, then the progress bar is unnecessary. If you believe that respondents might attempt to type text that is longer than the field width then a progress bar is desirable.

```
LargeTextBoxProgressBar=No
```

## **Fonts & Colors**

There are numerous commands to insert text into a survey page. These are:

Heading=This is a large heading.

Title=This is a title with smaller text.

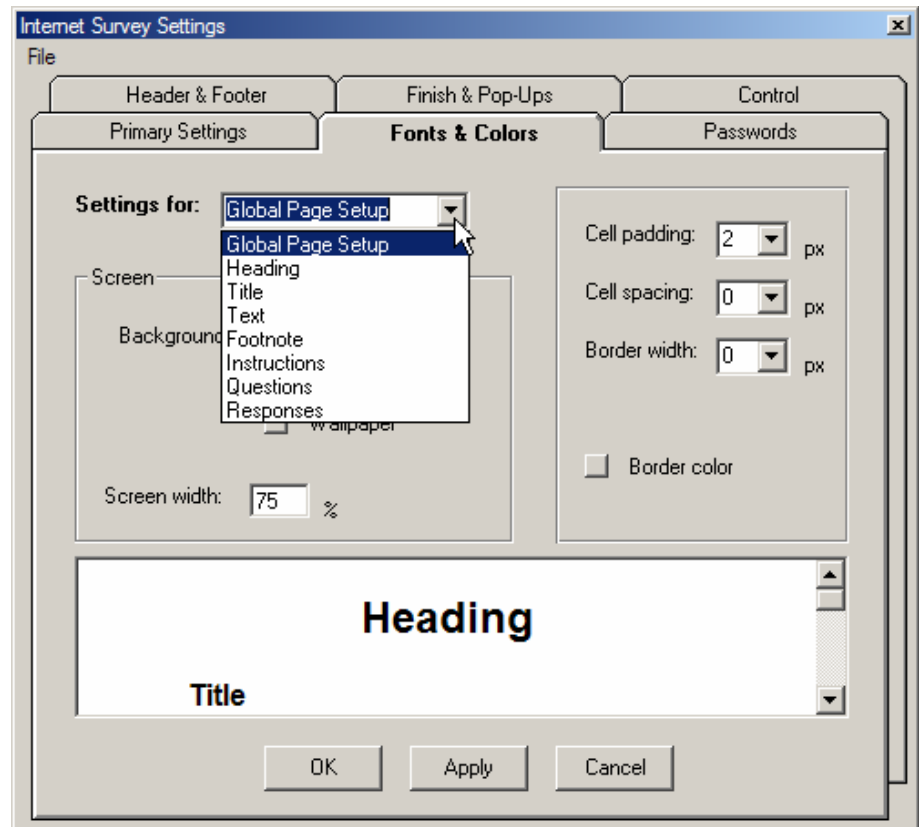
Text=This is normal text.

Footnote=This is the footnote at the bottom of the page.

Instructions=This is text in a frame.

Question=This is more text in a frame.

The attributes for these commands control the size, color, spacing, justification, and features like bold, underline and italics.



### ***Global Attributes***

The Global Attributes specify the default justifications, colors, and table attributes for the various components of the survey. The first letter of each attribute indicates that it is a global command ("G" stands for global), and applies to the web page as a whole.

GJustification=Center

GColor=#000000

GBgColor=#FFFFFF

GBorder=0

```
GCellpadding=2
GCellspacing=0
GWidth=75%
GBorderColor=#C0C0C0
Wallpaper=c:\StatPac\Wallpaper\background.jpg
```

GJustification sets the global justification. (Left, Center, or Right)

GColor sets the global font color. To select a color while editing the script, right click on the command line.

GBgColor sets the background color when a wallpaper is not specified. To select a color while editing the script, right click on the command line.

GBorder sets the global border thickness in pixels.

GCellpadding sets the space between the global frame and table cells in pixels.

GCellspacing sets the amount of space between the contents of a cell and the cell wall in pixels.

GWidth sets the survey width as a percent of the total screen width in percent.

GBorderColor sets the global border color when there is a border. To select a color while editing the script, right click on the command line.

Wallpaper sets the background wallpaper for the survey. To select the wallpaper, right click on the command line.

### ***Heading, Title, Text, & Footnote Attributes***

There are four commands that can be used to insert text into the survey without a frame (i.e., the global background color or wallpaper will appear as the background behind the text). These are the Heading, Title, Text and Footnote commands.

Each of these has its own attributes. Generally, heading is the largest font and footnote is the smallest.

#### ***----- Heading Attributes -----***

```
HeadingFontFace=Arial
HeadingFontSize=18
HeadingBold=Yes
HeadingUnderline=No
HeadingItalics=No
HeadingJustification=center
HeadingColor=#000000
```

#### ***----- Title Attributes -----***

```
TitleFontFace=Arial
TitleFontSize=12
TitleBold=Yes
TitleUnderline=No
TitleItalics=No
```

TitleJustification=left  
TitleColor=#000000

----- Text Attributes -----

TextFontFace=Arial  
TextFontSize=10  
TextBold=No  
TextUnderline=No  
TextItalics=No  
TextJustification=left  
TextColor=#000000

----- Footnote Attributes -----

FootnoteFontFace=Arial  
FootnoteFontSize=8  
FootnoteBold=No  
FootnoteUnderline=No  
FootnoteItalics=No  
FootnoteJustification=center  
FootnoteColor=#000000

\_FontFace sets the font face. (Any font name)

\_FontSize sets the font size. (Font size in points)

\_Bold sets whether the text will be bold. (Yes or No)

\_Underline sets whether the text will be underline. (Yes or No)

\_Italics sets whether the Heading will be italics. (Yes or No)

\_Justification sets the justification for the text. (Left, Center, or Right)

\_Color sets the font color for the text. To select a color while editing the script, right click on the command line.

### ***Instructions, Question, and Response Attributes***

There are three kinds of framed text. This text will appear in a frame with a border. The background color of the frame may be different from the global background color. Two of these (Instructions and Question) may be inserted into the page by using the command.

Instructions=Please answer the following items.

Question=Select you level of agreement or disagreement with the following:

The response attributes controls the appearance of the response choice frame. That is, the frame that displays the response choices and contains radio buttons, check boxes, or text boxes.

The first letter of the command indicates which type of attribute is being modified.

"I" stands for instructions, which is special text that can be shown to the user. The instruction attributes control the appearance of the instruction text.

"Q" stands for the question itself (i.e., the variable label). The question attributes control the appearance of the question text.

"R" is for the response categories or response text (i.e., the value labels). The response attributes control the appearance of the response choices.

There are several attributes for each of the framed components. These lines describe the attributes that will be used to create the Internet survey. Once you have set your preferences, they will rarely need to be changed in the script commands.

----- Instruction Attributes -----

```
IFontFace=Arial
IFontSize=10
IBold=Yes
IUnderline=No
IItalics=No
IJustification=left
IColor=#000000
IBgColor=#FFFFFF
IBorder=0
ICellpadding=2
ICellspacing=0
IWidth=100%
IBorderColor=#C0C0C0
```

----- Question Attributes -----

```
QFontFace=Arial
QFontSize=10
QBold=Yes
QUnderline=No
QItalics=No
QJustification=left
QColor=#000000
QBgColor=#DDDDDD
QBorder=1
QCellpadding=2
QCellspacing=0
QWidth=100%
QBorderColor=#C0C0C0
```

----- Response Attributes -----

RFontFace=Arial  
RFontSize=10  
RBold=No  
RUnderline=No  
RItalics=No  
RJustification=left  
RColor=#000000  
RBgColor=#FFFFFF  
RBorder=1  
RCellpadding=2  
RCellspacing=0  
RWidth=100%  
RBorderColor=#C0C0C0  
RBarColor=#F0F0F0

\_FontFace sets the font face. (Any font name)

\_FontSize sets the font size. (Font size in points)

\_Bold sets whether the text will be bold. (Yes or No)

\_Underline sets whether the text will be underline. (Yes or No)

\_Italics sets whether the Heading will be italics. (Yes or No)

\_Justification sets the justification for the text. (Left, Center, or Right)

\_Color sets the font color for the text. To select a color while editing the script, right click on the command line.

\_BGColor sets the background color of the frame. To select a color while editing the script, right click on the command line.

\_Border sets the size of the border around the frame in pixels.

\_Cellpadding sets the space between the global border and the frame border in pixels. *\_Cellpadding* is used to control the amount of space that the text will be indented. For example, if *RCellpadding=0* then the object for the response choices (radio buttons, check boxes, etc.) will be flush left. If *RCellpadding=5* then the object for the response choices will be indented five spaces (characters). Here are two examples. On the first one, *RCellpadding=3* and on the second, *RCellpadding=10*.

1. What is your gender?

- ☐ Male
- ☐ Female

1. What is your gender?

- ☐ Male
- ☐ Female

\_Cellspacing sets the space between the frame border and the text in the questions cell in pixels.

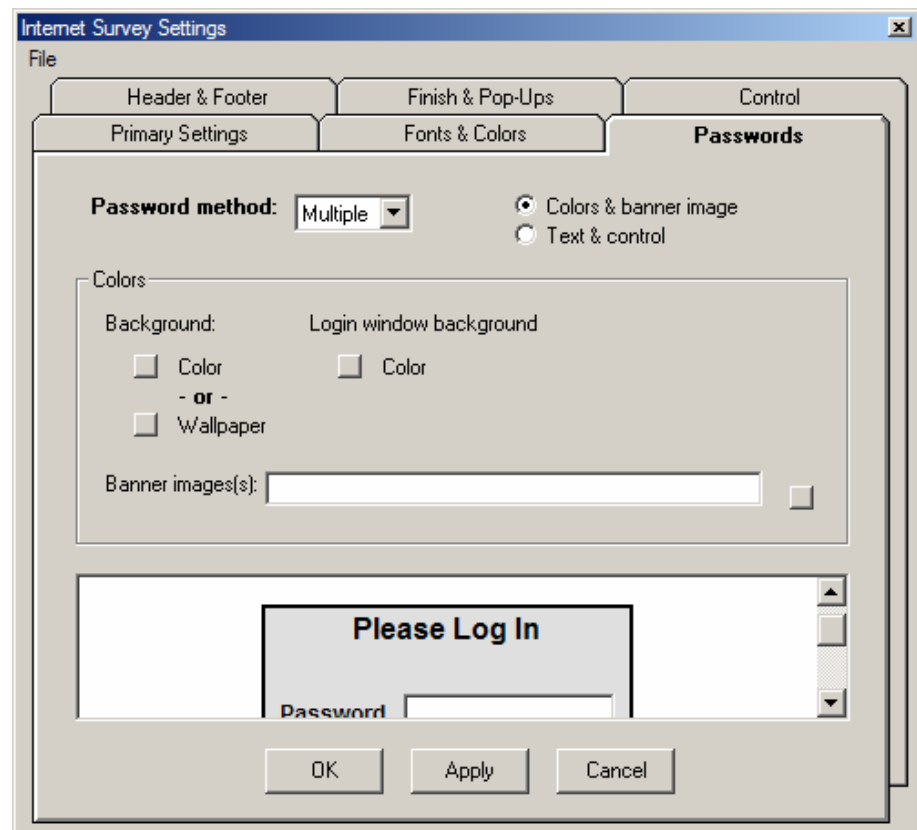
\_Width sets the width of the frame as a percent of the global width (usually 100%).

\_BorderColor sets the questions border color when there is a border.

RBarColor sets the background color of every other row in matrix style response choices.

## Passwords - Color & Banner Image

Password protected surveys are one way to control who has access to the survey.



### ***LoginBannerImage***

Sets the image that will appear at the top of the login page. To select the image while editing the script, right click on the command line.

`LoginBannerImage=c:\images\logo.jpg`

### ***LoginBGColor***

Sets the background color for the login page when no login wallpaper is specified. To select a color while editing the script, right click on the command line.

`LoginBgColor=#FFFFFF`



## LoginWallpaper

Sets the background wallpaper for the login page. To select the wallpaper while editing the script, right click on the command line.

LoginWallpaper=c:\images\background.jpg

## LoginWindowColor

Sets the background cell color for the login window. To select a color while editing the script, right click on the command line.

LoginWindowColor=#FFFFFF

## Passwords - Text & Control

The screenshot shows the 'Internet Survey Settings' dialog box with the 'Passwords' tab selected. The 'Password method' is set to 'Multiple'. The 'Text & control' section is expanded, showing fields for 'Heading' (Please Log In), 'Password prompt' (Password), 'Login button text' (Log In), 'Login failed text' (Invalid Login Attempt to), and 'Failed button text' (Try Again). There are also checkboxes for 'Show finish link', 'Email me', and 'Keep log', all set to 'No'. At the bottom, there are fields for 'Password field' and 'ID field', both set to '0'. The 'OK', 'Apply', and 'Cancel' buttons are at the bottom right.

## PasswordType

PasswordType sets the method that will be used for password protection. PasswordType may be set to type to None, Single, or Multiple.

When *PasswordType=None*, there will be no password protection. None of the other parameters in the Password Section need to be set. They are only important if *PasswordType* is set to *Single* or *Multiple*.

When *PasswordType=Single*, there will be one password for all respondents who access the survey.

When *PasswordType=Multiple*, each person who accesses the survey will have a unique password.

*PasswordType=Multiple*

### **LoginText**

LoginText sets the text shown at the top of the login window. It is the heading for the login window.

*LoginText=Please Log In*

### **PasswordText**

PasswordText sets the text on the login screen that prompts the respondent for the password. It will appear to the left of the textbox where a respondent enter their password.

*PasswordText=Password*

### **LoginButtonText**

LoginButtonText sets the text for the button on the login screen.

*LoginButtonText=Log In*

### **FailText**

FailText sets the message that will be shown to a respondent who enters an invalid password.

*FailText=Invalid Login Attempt to*

### **FailButtonText**

Sets the text on the retry button that will be shown to a respondent who enters an invalid password.

*FailButtonText=Try Again*

### **ShowLink**

ShowLink sets whether the finish page hyperlink (usually a link to your home page) will also be displayed on the login page. ShowLink may be set to Yes or No. The Primary settings of FinishText and FinishURL control the creation of the link.

*ShowLink=Yes*

### **EmailMe**

EmailMe sets the conditions when you will be emailed a notification of a login. EmailMe may be set to None, Valid, Invalid, or Both.

EmailMe=None No e-mail will be sent to you for either valid or invalid login attempts.

EmailMe=Valid An email will be sent to you when there has been a valid login.

EmailMe=Invalid An email will be sent to you when there has been an unsuccessful attempt to login.

EmailMe=Both An email will be sent to you when there has been a valid or invalid login attempt.

EmailMe=No

## **KeepLog**

KeepLog sets what kind of access will be saved in a server log. KeepLog may be set to None, Valid, Invalid, or Both. The actual log file will be named *codebookname.log* and will be stored in the same folder as the script (usually the cgi-bin folder). This is an ASCII text tab delimited file and may be downloaded and examined with any editor.

KeepLog=None No messages will be written to the server log.

KeepLog=Valid A message will be written to the server log when there has been a valid login.

KeepLog=Invalid A message will be written to the server log when there has been an unsuccessful attempt to login.

KeepLog=Both A message will be written to the server log when there has been an valid or invalid attempt to login.

KeepLog=Both

## **Passwords - Single vs. Multiple**

The script editor will ask for slightly different information depending on the password method (single or multiple). The link to a password protected survey is the same as a non-password protected survey.

### ***Password (single password method)***

Password sets the password for the survey. The same password is used for all respondents. The password can be any combination of letters and numbers. It is not case sensitive. We recommend short numeric passwords consisting of three to five digits or simple words or acronyms. Access to the survey will be limited to people knowing the password.

Password=Secret

### ***PasswordFile (multiple passwords method)***

PasswordFile sets the name of the file containing the passwords. Multiple password surveys use a data base of passwords stored on the server in the cgi-bin folder. The password file is in tab delimited ASCII text format. It must contain at least one field (the valid passwords), but it may contain other fields as well. It is often the same file that used to send e-mail invitations to participate in the survey. When a respondent

types a password at the login screen, the password file associated with the survey is examined to see if there is a match. To select a password file, right click on the command line.

```
PasswordFile=c:\survey\A83passwords.txt
```

### ***PasswordField & ID Field (multiple passwords method)***

Suppose you had an Excel file of email addresses and employee ID numbers. The first thing you would do is use Excel to write the two fields to a tab delimited file. The file would now look like this:

```
john@somedomain.com 465-35  
mary@anotherdomain.com 476-57
```

If you wanted to use employee ID as the password for the login, you would set PasswordField=2. Employee ID would be used as the password.

```
PasswordField=2
```

If you want to be able to track who responded and who didn't then you need to also specify an IDField. IDField identifies what field will be used as StatPac's internal RespondentID to identify that respondent. In the above example, if you wanted to use Employee ID to track respondents, then you would also set IDField=2.

```
PasswordField=2  
IDField=2
```

If you wanted to use employee ID as the password for the login but keep the survey anonymous, you would set PasswordField=2 and leave the IDField blank. When IDField is left blank, a random ID number will be generated when a respondent begins taking the survey. This will not allow you to track who responded and who didn't. It is the only method to create a password protected anonymous survey.

```
PasswordField=2  
IDField=
```

You could use the e-mail list management programs to add a random ID number to the tab delimited file for tracking respondents. StatPac always adds the ID number as the second field, so after running the program to add an ID number, the file might look like this:

```
john@somedomain.com    9867423  465-35  
mary@anotherdomain.com 3279684  476-57
```

You would set PasswordField=3 and set IDField=2. Then, respondents would enter their Employee ID number as the password and a random ID number would be used to track respondents. Note that this method is not anonymous because the random ID number could be used to link a particular respondent with their responses to the survey.

```
PasswordFile=c:\office\employee.txt  
PasswordField=3  
IDField=2
```

## ***PasswordControl***

PasswordControl sets the progress control that will be used on password protected surveys. It is only available for multiple password surveys and can be set to None, Once, or Forward.

Once means that a respondent can complete the survey only once. They can log in multiple times (always beginning on page one), but after they've reached the thank you page, they will not be able to log in again.

Forward means that they can log in multiple times, but they will always begin on the page where they left off. After they've reached the thank you page, they will not be able to log in again. Forward control only applies to multiple page surveys.

```
PasswordControl=Once
```

## **Passwords - Technical Notes**

The information in this section is not required to run StatPac. StatPac's Auto Transfer feature makes it unnecessary to know what files need to be uploaded to what folders. However, the information is important should you decide to manually upload or download files.

When you create a password protected survey StatPac does numerous things "behind the scenes".

For a single password method, a file called *studyname.text* is created in the project folder. It contains a single line consisting of the password. For a multiple password method, the password file you specify is written to the project folder using the name *studyname.text*. Auto Transfer uploads this file to the cgi-bin folder.

A file called *password.pl* will be created in your project folder. Auto Transfer uploads the file to the cgi-bin folder on your server and sets the permissions to 755. This is the program that reads the data base to determine if the respondent has entered a valid password.

The first page of the survey (*studyname.htm*) is renamed to *studyname\_start.htm* and the login page is named *studyname.htm*. Auto Transfer uploads *studyname\_start.htm*,

## Server Overrides

When using your own server, you can override your server settings by using these commands. We do not recommend the use of these commands since the appropriate way to make a change is to change your server settings. To use these commands, remove the leading ~ so the command is left justified in the script.

### **ActionTag**

ActionTag sets the full URL to StatPac's Perl script. The ActionTag setting will become part of the action tag in the html source code for the survey. The function of the Perl script is to take the answers from the html page and store them in a file on the server or email them to you. When a respondent clicks the submit button, it will direct the process to the location of the Perl script on the server.

The name of the Perl script on your local computer is statpac11.pl. Respondents will see the ActionTag setting in the address bar of their browser while taking the survey. For example, they would see:

`http://www.yourdomain.com/cgi-bin/statpac11.pl`

Therefore, the default script will change the name of statpac11.pl to your *surveyname.pl*. For example, if you are doing a survey for XYZ company, you might name the html files "XYZ". The default ActionTag setting would be:

`ActionTag=http://www.yourdomain.com/cgi-bin/XYZ.pl`

That way, respondents will see the company name in their address bar while they are taking the survey. On most servers, the www is optional and could be:

`ActionTag=http://yourdomain.com/cgi-bin/XYZ.pl`

If you are running on a secure server with SSL and your WebFolderURL uses a https:// prefix, then the ActionTag should also use https://

`ActionTag=https://yourdomain.com/cgi-bin/XYZ.pl`

You may name the Perl file to anything you want (except it must have a .pl extension). If you manually upload the files to your server, then you need to manually rename the file on your server from statpac11.pl to XYZ.pl. The important thing is that the name of the .pl file on your server is exactly the same as the ActionTag setting in the script.

Unix/Linux servers are case sensitive, so make sure you use the same case for the file name that you plan to give to respondents.

### **StorageFolder**

StorageFolder sets the path to the folder where responses will be stored on the server. Typically, responses are stored in the cgi-bin folder because it is not readily

accessible to the outside world. When left blank, StatPac will use the cgi-bin folder to store responses.

The actual file name for storing responses will be *codebookname.asc*.

The setting is specified differently for Unix and Windows servers.

### **Unix or Linux Server**

There are numerous ways to specify the storage folder on a Unix or Linux server. The setting may be specified as absolute or relative to the cgi-bin folder.

All of these would store the results in the cgi-bin folder.

```
StorageFolder=/home/username/public_html/cgi-bin
StorageFolder=../cgi-bin    (two periods)
StorageFolder=./            (one period)
```

All of these would store the results in a folder called "storage" which is immediately below the cgi-bin folder.

```
StorageFolder=/home/username/public_html/cgi-bin/storage
StorageFolder=../cgi-bin/storage    (two periods)
StorageFolder=./storage             (one period)
```

This would store the results in a folder called "private" which is at the same level as public\_html (and therefore not accessible to the outside world).

```
StorageFolder=../../private    (two periods)
```

These would store the results file in a folder called "storage" which is immediately below the public\_html folder. Note that this may pose a security risk because the storage folder would be accessible to the world.

```
StorageFolder=/home/username/public_html/storage
StorageFolder=./storage    (two periods)
```

### **NT, Windows, or IIS Server**

On an NT or IIS server the setting is specified using a DOS path (i.e., the full path beginning with a drive letter to the folder where responses should be stored). Users must have read/write access to the folder. An example might be:

```
StorageFolder=c:\inetpub\wwwroot\cgi-bin
```

In order to use FTP to retrieve the data, the wwwroot folder name must part of the StorageFolder path. For example, you would not be able to use FTP to retrieve this data because the storage folder is not below the wwwroot folder.

StorageFolder=d:\datastorage

Some IIS servers require that you use double backslashes instead of single backslashes. If you receive a “Server Busy” message after clicking the submit button on a survey, try changing the path to double backslashes in place of single backslashes:

StorageFolder=c:\\inetpub\\wwwroot\\cgi-bin

### ***ScriptFolder***

ScriptFolder sets the full URL of the cgi-bin folder (including the trailing /).

ScriptFolder=http://www.yourdomain.com/cgi-bin/

### ***Perl***

Perl is set to the absolute path where Perl is installed on your server. Your ISP will be able to tell you this information. The syntax for this setting is different for Unix/Linux and NT/IIS servers.

#### **Unix or Linux server:**

Perl=/usr/bin/perl

#### **Windows (NT or IIS) server:**

Perl=c:\\perl\\bin\\perl.exe

### ***MailProgram***

MailProgram is set to point to the mail program on your server. The MailProgram parameter only needs to be set if you are using password protection and want e-mail notification of login activity.

MailProgram=/usr/sbin/sendmail

---

## **Branching and Piping**

*Branching* is set up in the codebook using a semi-colon to indicate a skip pattern. Any skip patterns specified in the codebook will work for internet surveys. Nothing special needs to be done. Branching is supported for radio buttons.

A branch to another question on the same page will be immediate. As soon as the respondent clicks one of the radio buttons, the screen will scroll to the branched variable. A branch to a variable on another page may be immediate (when the



respondent clicks the radio button) or delayed until the respondent clicks the submit button.

The BranchDelay command may be used multiple times in the script to set the branching to immediate or delayed. It may be set to Yes or No. For example, the following command would delay the branching to another page until the submit button is clicked.

BranchDelay=Yes

*Piping* refers to displaying an answer from a previous question to a subsequent question. Piping is supported for multiple page surveys. You may pipe a response from one page to another page, but not to the current page.

For variables that contain value labels, the piped text will be the value label of the selected response choice. For variables that do not contain value labels, the piped text will be the actual text entered by the respondent. A piped response is requested on a subsequent page using the (\*VariableName\*) syntax in either an Instruction command, Question command, a variable label or value label in the codebook.

For example, suppose you had a variable named Gender with value labels of 1=male and 2=female. You also have a two open ended variables named FirstName and Age. The script for the survey might be:

TextBox FirstName

BlankLine

Radio Gender

BlankLine

TextBox Age

NewPage

Instructions=Hello (\*FirstName\*)! Thanks for taking our survey.

Instructions=You indicated that you are a (\*Age\*) year old (\*Gender\*).

BlankLine

Question=How would most (\*Gender\*) s respond to these items?

Radio V25-V30

Piping may also be used to prefill a textbox with a response to a question from a previous page. In order to prefill a textbox, you must use the ;(\*VariableName\*) syntax as a value label in the codebook. Note that the syntax begins with a semicolon when it is specified as a value label. For example, if a previous question had asked for “YourName”, and a question on a subsequent page asked for the Name of the CEO, you could prefill the Name of the CEO textbox with “YourName”, by adding a value label to the Name of the CEO variable in the codebook. The value label would be ;(\*YourName\*) on a line by itself.

---

## Randomization (Rotations)

StatPac supports value label randomization for radio buttons and checkboxes. It also supports variable randomization when groups of variables are displayed horizontally.

Traditional paper and pencil questionnaires typically use rotations for nominal data in order to vary the order in which response choices are presented to respondents. This is done to reduce bias that might be introduced by the order in which choices are listed on the page. Internet surveys have an advantage because they can randomize (not rotate) the order of the value labels (i.e., response choices).

The /R switch can be added to the end of the Radio or CheckBox commands to randomize the order that the response choices will be displayed in the respondent's browser. Each respondent will see a unique sequence of the response choices. For example, the following two commands would randomized the response choices for the variable 1 radio buttons and the multiple response variables 2 through 6. Use a space to separate the /R from the rest of the command.

**Radio V1 /R**

**CheckBox V2-V6 /R**

When used in the above form, all of the value labels will be randomized. This is sometimes undesirable when the last value label is an "other" response. Typically, you would want to leave the "other" choice as the last one displayed even though the other choices are randomized.

Add a number suffix to the /R switch to tell how many value labels should be excluded from the randomization. For example, if the last two value labels for variable 1 were "other" and "don't know" choices, the following command would exclude both value labels from the randomization (i.e., they will always be displayed as the last two choices).

**Radio V1 /R2**

In a similar example, the last value label would not be included in the randomization.

**CheckBox V2-V6 /R1**

The following would randomize the order of the variables in a series of horizontal radio buttons.

**Radio Vx- Vy /R**

Randomization will only work when the survey is online. It will not work on your local computer and the variable and value labels will always appear in the order specified in the codebook.

---

## Survey Creation Script

Most text in the survey will be generated from the codebook. There are many situations though where you will want to insert additional text into the survey (e.g., to give a section a heading).

### Overview

The //// SURVEY CREATION //// line is inserted as part of the initial script as a comment to let you know where the portion of the script begins that controls the variables. **Do not delete or change the location of this line.** It is required for StatPac to properly build the HTML survey pages.

Blank lines in the script are ignored. In long and complex surveys, you can add blank lines to the script to make it easier for you to keep track of pages. You can also begin any line with an apostrophe to make it a comment.

```
'Start the survey
Text=Thank you for your participation.
BlankLine
Text= No individual firms are identified and only aggregate data are made
public.
BlankLine

Title=Demographics
Radio Age
BlankLine
Radio Gender
BlankLine

NewPage

'Begin second page
Title=Attitudes
Question=Please rate the importance of each of the following items.
Radio Opinion_1 - Opinion_5

NewPage

'Begin third page
TextBox Q7
```

### ***Using Commands More than Once in a Script***

The Primary settings are only specified once at the beginning of the script. Most Advanced settings can be specified repeatedly in the Survey Creation portion of the script to change text, fonts, justifications, colors, and table attributes for the various questions of the survey.

For example, you could begin a survey with one color scheme and change to a different color scheme midway through the survey. To change an attribute, insert the command in the script at the place where you want the new value to take effect.

## **Specify Text**

There are 6 ways to specify text.

### ***Heading***

Heading is used to insert text using the heading attributes (usually very large font). There will be no frame around the text.

Heading=DEMOGRAPHIC SECTION

### ***Title***

Title is used to insert text using the title attributes. There will be no frame around the text.

Title=About You

### ***Text***

Text is used to insert text using the text attributes. There will be no frame around the text.

Text=Thank you for your participation. <br> <br> The information  
will be reported in aggregate and no individual will be identified.

### ***FootnoteText***

FootnoteText sets the text at the bottom of the page using the footnote attributes. Footnotes will always appear at the bottom of the page regardless of where you insert the command in the script.

FootnoteText=Sponsored by StatPac Inc.

### ***Instructions***

Instructions displays text in a frame using the instructions attributes.

Instructions=Please answer all items.

### ***Question***

Question displays text in a frame using the question attributes. It is most often used to add the question to grouped text boxes or a series of matrix style radio buttons.

Question=Please rate the importance of each of the following items.  
Radio Opinion\_1 - Opinion\_5

Question=Please rank the three most important items from this list by  
typing a 1, 2, and 3.  
TextBox Items\_1 - Items\_10

## **Spacing and pagination**

## ***BlankLine***

The *BlankLine* command has no equals symbol or parameters. It is inserted into the script at desired locations to create a blank line on the HTML page. It is usually inserted between each question.

```
Radio Age
BlankLine
Radio Gender
BlankLine
```

## ***NewPage***

The *NewPage* command has no equals symbol or parameters. Insert this line into the script when you want to end the current survey page and begin a new survey page. Each time the *NewPage* command appears, StatPac will create a new HTML page. StatPac's default script for a survey is for the entire survey to appear on a single page. If you want to create a multi-page survey, you'll have to manually insert the *NewPage* commands where you want to begin a new page.

```
NewPage
```

## **Images and Links**

### ***Image***

The *Image* command may be used to insert one or more graphics images into the survey. Type *Image=* and then right click to select the desired graphics file (.gif, .jpg, or .bmp). The format for the command is:

```
Image=c:\StatPac\Data\Picture.jpg
```

You can place multiple images side by side by specifying multiple filenames separated by semicolons.

```
Image=Filename;Filename;Filename
```

After the HTML is generated, you can use an HTML editor to manually insert a graphic, you must use either the fully qualified URL as the source file path or the special filename prefix of dot slash dot slash dot slash (i.e., *./././*) For example, either of the following would be acceptable methods of specifying a graphics source file:

```


```

When manually inserting a graphic with an HTML editor and you plan to use Auto Transfer to upload your files, the graphic must be in the project folder.

## Link

Inserts a hyperlink into the survey. It may be added to the script as a line by itself or may be added as a suffix to other commands.

Before using the Link command, first set the hyperlink information with the LinkURL and LinkText commands.

LinkURL=http://www.statpac.com

LinkText=Click here to see our home page

Then insert the Link command (without the equals symbol) where you would like the link to appear. It may be inserted on a line by itself in the script or may be added as a suffix to other commands. Each of the following would be correct uses of the Link command, and each would create a blue hyperlink on the survey page that a respondent could click on

Instructions=Please answer these questions. Link

Question=Please rate each of the following items: Link

Radio Q1 Link

DropDown Choices Link

Textbox Opinion Link

ListBox V12 Link

The Link commands may be used to make it easier for the respondent to email you. For example, if you have done an anonymous survey and you want respondents to be able to request a copy of the final report without any chance of linking their e-mail address to their responses, you could insert these three commands into the script:

LinkURL=mailto:admin@statpac.com?subject=Please send the report

LinkText=Click here to send an e-mail requesting a copy of the report

Link

## Help Windows

HelpWindow inserts a hyperlink into the survey for a popup help window. That is, it lets respondent to see additional text that does not appear on the survey itself (e.g., a help screen). The command may be added to the script as a line by itself or may be added as a suffix to other commands.

The actual appearance of the link created by the HelpWindow command and the contents of the popup window are controlled by three parameters. These parameters should be set in the script before using the HelpWindow command. When the HelpWindow command is executed, it will use the current settings of the three parameters. The three parameters are:

HelpLinkText=Text

HelpText=Text

HelpFileName=Filename

HelpLinkText is the blue link text that the respondent will see on the survey. If not specified, the default is "Help".

HelpText is the actual text that will appear in the popup window when opened. It may use imbedded HTML tags to control fonts and spacing.

```
HelpLinkText=Please click here for more information
HelpText=This additional information is provided to help you answer the
question<br>Please try to be objective answering this question. It is
important that you are truthful and give your best answer.
HelpWindow
```

If you already have an .htm file that you want to use as the contents of the popup window, then you would not use the HelpText command. Instead, use the HelpFileName command to specify the name of the existing .htm file. If you plan to use Auto Transfer to upload your files, this file must be in the project folder.

```
HelpLinkText=Click here for an explanation
HelpText=
HelpFileName=c:\StatPac\Data\explanation.htm
HelpWindow
```

After setting the parameters, the HelpWindow command. Actually creates the window. The HelpWindow command may appear on a line by itself in the script or may be added as a suffix to other commands. For example, each of the following would be correct uses of the HelpWindow command, and each would create a link on the survey page that a respondent could click on to evoke the help window:

```
HelpWindow
Instructions=Please answer these questions. HelpWindow
Question=Please rate each of the following items: HelpWindow
Radio V5 HelpWindow
DropDown V9 HelpWindow
Textbox V45 HelpWindow
ListBox V12 HelpWindow
```

The names of the popup window pages will be "surveyname\_help1.htm", "surveyname\_help2.htm", etc. For each occurrence of the HelpWindow command in the script, a new popup page will be created. The popup window pages are .htm files and must be uploaded to the server to the same folder as your survey.

Here is an example of a script that creates three links in the survey that a respondent could click on to evoke three different popup windows. It will also create three different help files.

```
HelpLinkText=Click Here for Information about our sponsor
```

```

HelpText=This survey is sponsored by StatPac Inc.<br>You can call us
at (715) 442-2261
HelpWindow
BlankLine
HelpLinkText=Our Privacy Policy
HelpText=We don't tell anyone anything!
Instructions=Please answer all questions HelpWindow
BlankLine
HelpLinkText=Help
HelpText=Please answer this question from the perspective of your average
customer. Try to answer how you think your typical customer would
respond.
Radio V1 HelpWindow

```

## Popup Windows

Popup windows provide another way of branching. They are useful when you want to ask (or not ask) a few additional questions depending on the respondent's answer to another question without changing pages from the respondent's perspective.

An example would be a question where you say, "If yes, please answer the following." In a multiple page survey, you would place the yes/no question on one page, the conditional question(s) on a second page, and the rest of the questions on subsequent pages. The popup window offers the same functionality, except the conditional questions will appear in a new (popup) window, and the rest of the questions are on the same html page as the yes/no question.

The actual branching instructions are set up in the codebook using the semicolon syntax. For example, suppose we have a codebook with a branch so people who drink coffee are asked questions 1a and 1b, while people who don't drink coffee are not asked those questions. All value labels immediately prior to the popup must specify a branch (even if it's just to the next variable, which is in the popup window).

Variable Label 1: 1. Do you drink coffee?

1=Yes ; 2

2=No ; 4

Variable Label 2: 1a. What is your favorite brand of coffee?

Variable Label 3: 1b. How often do you drink coffee?

Variable Label 4: 2. How old are you?

You can adjust the StatPac script so that questions 1a and 1b will appear on a popup window. The PopupStart and PopupEnd commands may be inserted into the script to create the popup window. In this example, the script might be:

```

Radio V1
BlankLine
PopupStart

```



TextBox V2  
BlankLine  
TextBox V3  
BlankLine  
PopupEnd  
TextBox V4

The surveyname\_1.htm file will have questions 1 and 4. The popup page surveyname\_popup1.htm file will have questions 2 and 3. The first page the respondent sees will have questions 1 and 2. They will begin by answering question 1. If they select “yes”, a popup window will appear “on top” of the page they are seeing. The popup window will show questions 1a and 1b. When they press the continue button on the popup window, it will close, and the respondent will see the main page again, showing questions 1 (already answered) and 2.

Popup windows are regular html pages. The only difference from a regular survey page is the way in which they are evoked. Popup windows will be named surveyname\_popup1.htm, surveyname\_popup2.htm, surveyname\_popup3.htm, etc.

There are several rules and limitations for using popup windows:

1. Popup windows will only work online. You may view the popup page on your local computer, but you cannot test the popup feature until the files are uploaded to a server.
2. You may branch within a popup window, but you may not branch out of a popup window. That is, you may not branch to a variable that is not in the popup window. Complex branching is not supported within a popup window.
3. Piping is not supported within a popup window. You cannot pipe to or from a popup window.
4. In the variable that evokes the popup window, all value labels must specify a branch, even if it's to the next variable (which will be in the popup window).

Popup windows are ideal for situations where you have a small number of conditional questions. When there are many conditional questions, use the NewPage command instead of a popup window.

---

## Survey Creation - Objects

The input methods (called objects) include radio buttons, drop down menus, text boxes, check boxes, and list boxes. When StatPac first creates the default script, it will select the objects that seem most appropriate to the questions. However, after the default script has been created, the objects can be changed by simply changing the Survey Creation portion of the script.

The following is a description of the commands that can be used to create objects. In all of commands, variable names can be used instead of the "V" numbers.

### Radio Buttons for a Single Variable

**Syntax: *Radio Vx -or- Radio (parameters) Vx***

The Radio command will create a radio button for each value label. It is most appropriate when there are a small number of choices. The first form of the command will create all the radio button in a single column. The second form of the command will allow you to specify various display parameters. The parameters are:

- V H vertical or horizontal
- 1 2 one or two columns of radio buttons (applies to vertical format only)
- A E label all points or label only the end points
- Y N show numeric codes (yes or no)

Parameters may be any combination in any order (upper or lower case). The following would create a group of horizontal radio buttons for variable nine with only the end points labeled and numeric codes above each radio button.

#### Radio (HEY) V9

In creating the default script, if there are six or fewer value labels, the default will be one column. If there are 7-12 value labels, the default will be two columns. For more than 12 value labels, the default will be a DropDown menu.

How old are you?	
<input type="radio"/> 14 years or younger	<input type="radio"/> 45 - 54
<input type="radio"/> 15 - 19	<input type="radio"/> 55 - 64
<input type="radio"/> 20 - 24	<input type="radio"/> 65 - 74
<input type="radio"/> 25 - 34	<input type="radio"/> 75 years & over
<input type="radio"/> 35 - 44	

When creating horizontal radio buttons (either single questions or a matrix), one or two minus signs can be included in the parameters to hide the codes for the one or two highest value labels. For example, suppose you have a 1-5 Likert scale and you also have 6=No Opinion. Including a minus sign in the parameters would still show No Opinion text, but hide the 6 code in the numeric scale. If the two highest response choices were 6=No Opinion and 7=Not Applicable, then two consecutive minus signs in the parameters would hide the numeric codes for both categories. This feature can be used for single variables in a horizontal format or matrix style radio buttons.

#### Radio (HAY--) V9

How would you rate the new feature?						
Very Good	Good	Fair	Poor	Very Poor	No Opinion	Did Not Use
1	2	3	4	5		
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Radio (-) Q5a - Q5f

BlankLine

5. Please rate StatPac on each of the following criteria.						
	Very Good	Good	Fair	Poor	Very Poor	No Opinion
	1	2	3	4	5	
5a. Overall, how would you rate the StatPac software?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
5b. Quality of the documentation.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
5c. Ease of learning.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
5d. Ease of use.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
5e. Power and completeness.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
5f. Quality of the technical support.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

## Radio Buttons for Grouped Variables (matrix style)

**Syntax: Radio Vx – Vy -or- Radio (parameters) Vx - Vy**

When your survey has a series of Likert scale or semantic differential scale items, a horizontal radio button format can be specified. Horizontally grouped radio buttons are often referred to as *matrix* questions.

The first form of the command will create all the radio button in a single column. The second form of the command will allow you to specify various display parameters. The parameters are:

L R radio buttons to left or right of text

A E label all points or label only the end points

Y N show numeric codes (yes or no)

C D describes codebook format as close or distant for two side-by-side radio button groups

The L and R parameters may be used to show the questions to the left or right of the radio buttons. The default can be set with the *RadioTextPosition* command.

The A and E parameters may be used to label all points or just the end points of the scale. The A parameter will label every response choice. The E parameter will label only the anchors of the scale.

The Y and N parameters control whether numeric codes are used to label the scale.

Parameters may be any combination in any order (upper or lower case). The following would create a group of horizontal radio buttons for variables Q5a to Q5f with text to the left of the buttons, only the end points labeled, and numeric codes above each radio button.

### Radio (LEY) Q5a - Q5f

You will need to manually edit the default script to display a series of items in this format. When you create the default script, StatPac doesn't know which items should be grouped together, so each item will be specified as an individual radio button variable. The default script might look like this:

```
Radio Q5a
BlankLine
Radio Q5b
BlankLine
Radio Q5c
BlankLine
Radio Q5d
BlankLine
Radio Q5e
BlankLine
Radio Q5f
BlankLine
```

To convert these to matrix format, you would change the script so the items to be grouped together are specified as a range on a single script line instead of each on their own script line.

```
Radio Q5a - Q5f
BlankLine
```

	Very Good			Very Poor	
	1	2	3	4	5
5a. Overall, how would you rate the StatPac software?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
5b. Quality of the documentation.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
5c. Ease of learning.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
5d. Ease of use.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
5e. Power and completeness.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
5f. Quality of the technical support.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

If you want to show a single variable in horizontal matrix style format, include the dash without the second variable, as in: Radio Vx- or Radio (parameters) Vx-.

The C (close) or D (distant) parameter should only be used when you want two groups of radio buttons to appear side by side. For example, you may have three

attribute variables. You want to ask respondents to rate each of the three attributes and you also want them to assign an importance rating to each attribute. You want the layout on the screen to look like this:

Attribute 1	Rating 1	Importance 1
Attribute 2	Rating 2	Importance 2
Attribute 3	Rating 3	Importance 3

You can set up the codebook two different ways. The C or D describes which way the codebook is set up.

In this codebook, the Rating and Importance variables are close to each other.

```
V1 Attribute 1 Rating
V2 Attribute 1 Importance
V3 Attribute 2 Rating
V4 Attribute 2 Importance
V5 Attribute 3 Rating
V6 Attribute 3 Importance
```

The command would be:

Radio (C) v1-v6

In this codebook, the Rating and Importance variables are distant from each other.

```
V1 Attribute 1 Rating
V2 Attribute 2 Rating
V3 Attribute 3 Rating
V4 Attribute 1 Importance
V5 Attribute 2 Importance
V6 Attribute 3 Importance
```

The command would be:

Radio (D) v1-v6

Here is an example for 4 attributes rated on quality and overall impression: The variables are:

```
V1 Attribute 1 – Overall Quality
V2 Attribute 2 – Overall Impression
V3 Attribute 3 - Documentation Quality
V4 Attribute 4 - Documentation Impression
```

- V5 Attribute 1 – Power Quality
- V6 Attribute 2 – Power Impression
- V7 Attribute 3 – Support Quality
- V8 Attribute 4 – Support Impression

The command to create the side-by-side radio button matrix would be:

Question=Please rank the quality and your overall impression of each of the following.

Radio (CN) V1-V8

BlankLine

Please rank the quality and your overall impression of each of the following.									
	Very Good	Good	Fair	Poor	Very Poor	Low	Medium	High	
Overall	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
Documentation	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
Power	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
Support	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	

A textbox can be placed next to the Other response category. When used with a side-by-side matrix, the Other variable must follow the last variable in the variable list. In both examples, V7 would be the Other variable.

## DropDown Menu

### **Syntax: DropDown Vx -or- DropDown (y) Vx**

The dropdown menu is used when there are a large number of possible categories for the respondent to choose from. The most common example of this is when asking for a country or state. The first form of the command will create a dropdown menu where only one line shows until the menu is selected by the user. You can set the default text for the line by inserting the command: DropDownDefault=some text. In the second form of the command, you can specify the number of value labels to display in the dropdown window before the dropdown occurs. Y is the number of lines you want to display before the user selects the menu. If y equals the number of value labels, then all the value labels will be shown all the time.

DropDown (5) Country

## TextBox for a Single Variable

**Syntax:** *TextBox Vx -or- TextBox (parameters) Vx*

A TextBox is used when requesting that the user type something rather than choose from a menu of items. The default script will use a textbox if there are no value labels for a variable and the variable has a field width greater than 1. When the field width for a variable is a long alpha field (greater than A60), the textbox will have multiple lines and a scroll bar. If you want to force a long alpha field to use a single line TextBox, then use the command format of: `TextBox (1) Vx`

The parameters for the textbox are:

- L C** left justify or center the TextBox in the frame
- n** number of lines to show in the TextBox (1 to 5)
- P N** progress bar or no progress bar (multiple line TextBox only)

The following TextBox was centered, shows five lines, and show no progress bar as the respondent is typing:

`TextBox (C5N) Q7`

StatPac has a special feature that lets you attach a textbox to the last item of a radio button variable, set of multiple response variables, and a series of horizontal radio

buttons. Here is an example of a textbox that has been attached to a radio button variable.

What is your favorite brand?	
<input type="radio"/> Brand X	
<input type="radio"/> Brand Y	
<input type="radio"/> Brand Z	
<input type="radio"/> Other	<input type="text"/>

To use this feature, simply create an alpha variable immediately following the variable (or variables) that you want to contain the text box. Use the word "Other" as the variable label for the alpha variable and do not specify the alpha variable in the StatPac script. The codebook for the above example would be:

Variable 1 Label: What is your favorite brand?

1=Brand X

2=Brand Y

3=Brand Z

4=Other

Variable 2 Label: Other

The two criteria necessary to use this feature are: 1) the alpha variable has a variable label of "Other", and 2) the StatPac script doesn't include the alpha variable. In the above example, the StatPac script would omit variable 2 (i.e., it would not include a command line: TextBox V2).

Here is an example of a textbox that has been attached to the last in a series of multiple response variables. There are four multiple response variables followed by a single alpha variable with a variable label of "Other".

Where did you hear about the product? (check all that apply)	
<input type="checkbox"/> Radio	
<input type="checkbox"/> TV	
<input type="checkbox"/> Newspaper	
<input type="checkbox"/> Other (specify)	<input type="text"/>

The codebook to produce this example would be:

Variable 1 Label: Where did you hear about the product (check all)

1=Radio

2=TV

3=Newspaper



4=Other (specify)

Variable 2 Label: Where did you hear about the product (check all)

1=Radio

2=TV

3=Newspaper

4=Other (specify)

Variable 3 Label: Where did you hear about the product (check all)

1=Radio

2=TV

3=Newspaper

4=Other (specify)

Variable 4 Label: Where did you hear about the product (check all)

1=Radio

2=TV

3=Newspaper

4=Other (specify)

Variable 5 Label: Other

Finally, here is an example of a textbox that has been attached to a group of horizontal radio buttons.

	Excellent		Poor	
	1	2	3	4
How would you rate the taste?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
How would you rate the smell?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
How would you rate the texture?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
How would you rate the appearance?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
How would you rate the package design?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Other (please specify)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

The last two variables in the codebook used to create this example are:

Variable 6 Label: Other (please specify)

1=Excellent

2=

3=

4=Poor

Variable 7 Label: Other

You can change the variable label used to trigger this feature by editing the StatPac.ini file. Search for *HTML\_OtherText = Other* and change it to the variable label you want to use to evoke this feature.

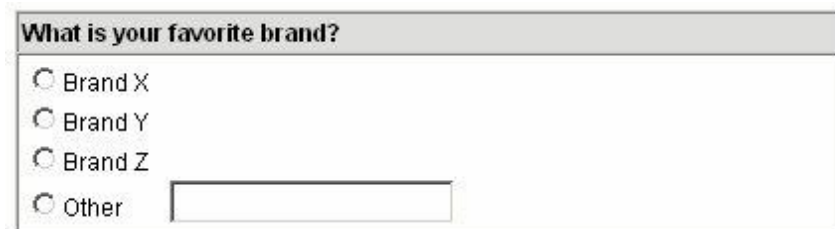
## Adding a TextBox to a Radio Button, CheckBox, or Radio Button Matrix

StatPac has a special feature that lets you attach one or more textboxes to a radio button variable, set of multiple response variables, or a series of horizontal radio buttons. There are two ways to do this depending on the number and position of the textboxes you want to add. In its simplest form, one textbox is added to the last item. In the more complex form, a textbox can be added to one or more radio buttons or checkboxes.

In all of the following examples, the textbox variables are specified as alpha format and should use a field length long enough to hold the longest expected comment. In these examples a format of A200 for the textbox variables would most likely be sufficient.

### Attaching One Textbox to the Last Item

StatPac has a special feature that lets you attach a textbox to the last item of a radio button variable, set of multiple response variables, and a series of horizontal radio buttons. Here is an example of a textbox that has been attached to a radio button variable.



What is your favorite brand?

☐ Brand X

☐ Brand Y

☐ Brand Z

☐ Other

To use this feature, simply create an alpha variable immediately following the variable (or variables) that you want to contain the text box. Use the word "Other" as the variable label for the alpha variable and do not specify the alpha variable in the StatPac script. The codebook for the above example would be:

Variable 1 Name: Brand

Variable 1 Label: What is your favorite brand?

1=Brand X

2=Brand Y

3=Brand Z

4=Other

Variable 2 Name: Other\_Brand\_Specified

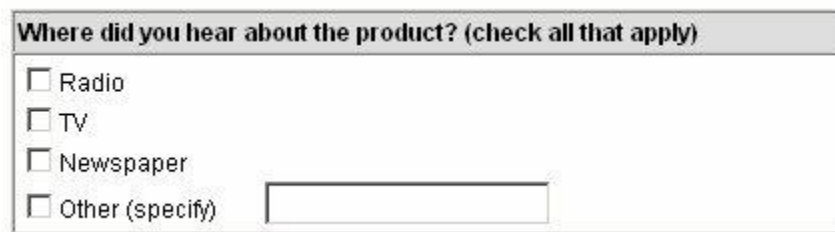
Variable 2 Label: Other

The two criteria necessary to use this feature are: 1) the alpha variable has a variable label of "Other", and 2) the StatPac script doesn't specify the alpha variable.

In the above example, the StatPac script would omit variable 2 (i.e., it would not include a command line: `TextBox Other_Brand_Specified`). The command to produce the above example would be:

**Radio Brand**  
**BlankLine**

Similarly, a textbox can be attached to the last in a series of multiple response variables. There are four multiple response variables followed by a single alpha variable with a variable label of "Other".



**Where did you hear about the product? (check all that apply)**

☐ Radio

☐ TV

☐ Newspaper

☒ Other (specify)

The codebook to produce this example would be:

Variable 1 Name: Hear\_1

Variable 1 Label: Where did you hear about the product (check all)

1=Radio

2=TV

3=Newspaper

4=Other (specify)

Variable 2 Name: Hear\_2

Variable 2 Label: Where did you hear about the product (check all)

1=Radio

2=TV

3=Newspaper

4=Other (specify)

Variable 3 Name: Hear\_3

Variable 3 Label: Where did you hear about the product (check all)

1=Radio

2=TV

3=Newspaper

4=Other (specify)

Variable 4 Name: Hear\_4

Variable 4 Label: Where did you hear about the product (check all)

1=Radio

2=TV

3=Newspaper  
 4=Other (specify)  
 Variable 5 Name: Hear\_Other\_Specified  
 Variable 5 Label: Other

In this example, the StatPac script would not specify variable 5. The command to produce the above example would be:

CheckBox Hear\_1 - Hear\_4  
 BlankLine

Finally, here is an example of a textbox that has been attached to the last variable in a horizontal radio button matrix.

		Excellent		Poor	
		1	2	3	4
How would you rate the taste?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
How would you rate the smell?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
How would you rate the texture?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
How would you rate the appearance?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
How would you rate the package design?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
Other (please specify)	<input type="text"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

The last two variables in the codebook used to create this example are:

Variable 6 Name: Other\_Rating  
 Variable 6 Label: Other (please specify)  
 1=Excellent  
 2=  
 3=  
 4=Poor  
 Variable 7 Name: Other\_Rating\_Specified  
 Variable 7 Label: Other

The StatPac script would not specify variable 7. The command to produce the above example would be:

Radio Taste - Other\_Rating  
 BlankLine

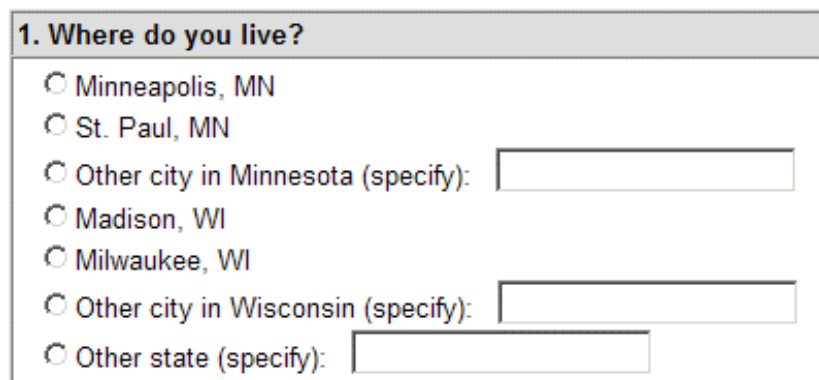
In all of the above examples, the variable label "Other" is used to evoke this feature. You can change the variable label used to trigger this feature by editing the StatPac.ini file. Search for *HTML\_OtherText = Other* and change it to the variable label you want to use to evoke this feature.

### ***Attaching More than One Textbox***

When you want to attach a textbox to more than one item in a radio button variable, set of multiple response variables, or a series of horizontal radio buttons, a different form of the syntax is required. This syntax can be used to attach a textbox to any of the items (not only the last item).

The variable name for the textbox variables is how you control which radio buttons will have textboxes. All textbox variable names end with "\_Other".

Here is an example of a radio button variable that has three textboxes attached to it.



The textbox items immediately follow the radio button variable. Their variable name has three parts with an underscore separating the parts: 1) the name of the radio button variable, 2) the value label code in the radio button variable, and 3) the word "Other".

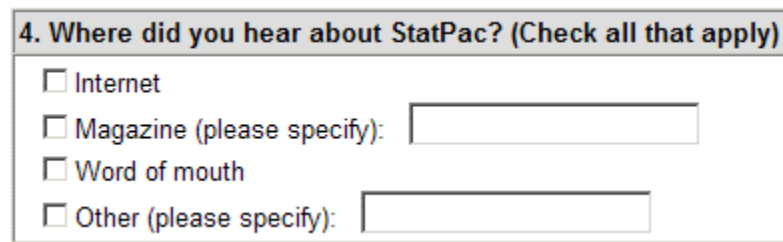
The codebook to produce the above example would look like this:

```
Variable 1 Name: Residence
Variable 1 Label: 1. Where do you live?
    1=Minneapolis, MN
    2=St. Paul, MN
    3=Other city in Minnesota (specify):
    4=Madison, WI
    5=Milwaukee, WI
    6=Other city in Wisconsin (specify):
    7=Other state (specify):
Variable 2 Name: Residence_3_Other
Variable 3 Name: Residence_6_Other
Variable 4 Name: Residence_7_Other
```

The variables for the textboxes are not specified in the StatPac script, So the script for this example would simply be:

Radio Residence  
BlankLine

More than one textbox can also be included in a set of multiple response variables. Here is an example where two textboxes are attached to a set of multiple response variables:



4. Where did you hear about StatPac? (Check all that apply)

☐ Internet

☐ Magazine (please specify):

☐ Word of mouth

☐ Other (please specify):

The textbox variables immediately follow the multiple response variables. The textbox variable names have two parts separated by an underscore: 1) the name of the multiple response variable, and 2) the word "Other".

Here is the codebook for the previous example:

Variable 4 Name: Hear\_1

Variable 4 Label: 4. Where did you hear about StatPac? (check all)

1=Internet

2=Magazine (please specify):

3=Word of mouth

4=Other (please specify)

Variable 5 Name: Hear\_2

Variable 5 Label: 4. Where did you hear about StatPac? (check all)

1=Internet

2=Magazine (please specify):

3=Word of mouth

4=Other (please specify)

Variable 6 Name: Hear\_3

Variable 6 Label: 4. Where did you hear about StatPac? (check all)

1=Internet

2=Magazine (please specify):

3=Word of mouth

4=Other (please specify)

Variable 7 Name: Hear\_4

Variable 7 Label: 4. Where did you hear about StatPac? (check all)

1=Internet

2=Magazine (please specify):

3=Word of mouth

4=Other (please specify)

Variable 8 Name: Hear\_2\_Other

Variable 8 Label (optional): Magazine specified

Variable 9 Name: Hear\_4\_Other

Variable 9 Label (optional): Other specified

The variables for the textboxes are not specified in the StatPac script, So the script for this example would be:

CheckBox Hear\_1 - Hear\_4

BlankLine

Textboxes can also be attached to any or all items in a horizontal radio button matrix. Here is an example where two textboxes are attached to all of the items in a radio button matrix:

Please rate our product on each of the following criteria. Feel free to add comments.					
	Very Good	Good	Fair	Poor	Very Poor
	1	2	3	4	5
Quality: <input type="text"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Service: <input type="text"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Value: <input type="text"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

The textbox variables immediately follow the last matrix variable. The textbox variable names have two parts separated by an underscore: 1) the name of the matrix variable, and 2) the word "Other".

Here is the codebook for the matrix example:

Variable 1 Name: Quality

Variable 1 Label: Quality:

1=Very Good

2=Good

3=Fair

4=Poor

5=Very Poor

Variable 2 Name: Service

Variable 2 Label: Service:

1=Very Good

2=Good

3=Fair

4=Poor  
 5=Very Poor  
 Variable 3 Name: Value  
 Variable 3 Label: Value:  
 1=Very Good  
 2=Good  
 3=Fair  
 4=Poor  
 5=Very Poor  
 Variable 4 Name: Quality\_Other  
 Variable 5 Name: Service\_Other  
 Variable 6 Name: Value\_Other

The variables for the textboxes are not specified in the StatPac script, So the script for this example would be:

Question=Please rate our product on each of the following criteria. Feel free  
 to add comments.  
 Radio Quality - Value  
 BlankLine

## TextBoxes for Grouped Variables

### **Syntax: *TextBox Vx - Vy -or- TextBox (parameters) Vx - Vy***

More than one TextBox can be inserted into a single frame by specifying a variable range. Fields with less than 40 characters will shown as a single line TextBox and longer fields will be shown as a multiple line text box with a scroll bar.

Parameters may be used to control the number of lines in each TextBox and the location of the text. Text may be placed to the left, right, top, or bottom of the TextBox using the LRTB parameters. The number of lines for the TextBox can also be specified. For example, in the following command, parameters were used to set the text to the right and to create single-line TextBoxes:

Question=Please rank each of the following by typing a 1 for the most  
 important, 2 for the second most important, and so on.  
 TextBox (R1) Taste - Package

Please rank each of the following by typing a 1 for the most important,  
 2 for the second most important, and so on.

<input type="text"/>	Taste
<input type="text"/>	Smell
<input type="text"/>	Texture
<input type="text"/>	Appearance
<input type="text"/>	Packaging



There are two special validity checks that can be applied to grouped text boxes. These checks can be added to the survey by inserting a line in the script.

### ***Constant Sum Validity Check for TextBoxes***

The first is a constant sum, where you want the responses to add up to a certain value. An example would be "What percent of your total time do you spend..." and you want the percents for each of the items to add to 100. The Sum command may be inserted anywhere in the script. The format of the command is: Sum (the desired sum) Vx – Vy. The desired sum is enclosed in parentheses, and Vx and Vy should be the same as the TextBox command. An example would be:

Question=What percent of your total time do you spend...  
TextBox Eating – Sleeping  
Sum (100) Eating – Sleeping

In Internet surveys it is very important not to frustrate respondents. Sometimes respondents' sums are off a little bit (e.g., they may add to 99 or 101 instead of 100). The /E switch may be added to the end of the Sum command to specify the amount of error that can be accepted before the respondent will get an error message. For example, if you were willing to accept sums that were between 98 and 102, the commands would be:

Question=What percent of your total time do you spend...  
TextBox Eating – Sleeping  
Sum (100) Eating – Sleeping /E2

A special /T switch may also be added to the end of the Sum command. This will add a disabled TextBox at the bottom labeled Total. It will be updated dynamically as respondents type numbers into the other fields of the TextBox Group. This feature is useful when you have asked respondents to make their responses total a given number because it lets them see their current total in real time and make adjustments to their responses to achieve the desired total.

Question=Make the following items total \$100  
TextBox Food - Insurance  
Sum (100) Food - Insurance /T

You may want to allow respondents to leave a series of constant sum TextBoxes blank. That is, respondents can leave the items blank or they can fill in the items so they add to a constant sum. The Missing Allowed field in the codebook controls this feature. When Missing Allowed is checked in the codebook for the first variable in the series (i.e., Vx), then the respondent can leave all the items blank or they can fill them in so they add to a constant sum. When Missing Allowed is not checked, they will be required to make the items add to a constant sum and will not have the option to leave them blank.

## Ranking Validity Check for TextBoxes

The second type of validity check that can be applied to grouped text boxes is for ranking questions. Respondents are often asked to rank items (e.g., type a 1 for the most important item, a 2 for the next most important item, etc.). The ranking validity check will check for duplicate or missing ranks. The Rank command may appear anywhere in the script. The syntax of the command is: Rank (number of ranks) Vx – Vy. The desired number of ranks is enclosed in parentheses, and Vx and Vy should be the same as the TextBox command. If the parentheses and desired number of ranks is omitted, all items in the list will need to be ranked by the respondent.

An example where you ask respondents to rank the top two items only would be:

Question=Rank the two most important items:

TextBox Salary - Benefits

Rank (2) Salary - Benefits

You might want to allow respondents to leave a series of ranking TextBoxes blank. That is, respondents can leave the items blank or they can rank the items. The Missing Allowed field in the codebook controls this feature. When Missing Allowed is checked in the codebook for the first variable in the series (i.e., Vx), then the respondent can leave all the items blank or they can rank them as specified by the Rank command. When Missing Allowed is not checked, they will be required to rank the items and will not have the option to leave them blank.

## Slider for Single or Multiple Variables

**Syntax: Slider Vx - Vy -or- Slider (parameters) Vx - Vy**

The Slider command is used to create one or more sliders with a scaled ruler and a handle that allows a respondent to visually set a value by moving the handle with the mouse. The general appearance of a slider is:



Labeling for the slider scale is controlled either by value labels or specified parameters. The slider will always show an adjacent TextBox to allow the respondent to manually type the response,

The Slider command may be used as an alternative way to present psychometric scaling items or a textbox requiring a numeric response. The slider will never be created as a default object. If you want to use a slider, you must modify the script by changing a Radio or Textbox command to a Slider command.

The Slider is appropriate for Likert and semantic differential scales (e.g. agree to disagree, very important to not important, favor to oppose, etc), These scales are usually 1-5, 1-7, or 1-10. An example might be:

Variable 2 Label: How important is the amendment?

1=Not Important

2=

3=  
4=  
5=*Very Important*

The default script would be: Radio V2  
You would change the script to be: Slider V2

The Slider is also appropriate for questions where you ask the respondent to rate one or more items:

For example: Please rate your level of optimism about the economy on a scale from 0 to 100. In the codebook, you would define the variable as N3 and set the value labels to 0-100.

The default script would be: TextBox V5  
You would change the script to be: Slider V5

Parameters can be used for labeling the slider. The parameters are separated from each other by commas. The format for the parameters is:

*(Title, Low Label, Middle Label, High Label)*

The title will appear above the scaled ruler. The low, middle, and high labels will be taken from the value labels or can be specified or changed with the parameters. Examples might be:

Example: Slider (Percent) V9  
(Create a slider for V9 with a title of Percent, and use the existing value labels to label the lowest, middle, and highest positions of the slider).

Example: Slider (Percent,Negative,Neutral,Positive) V10  
(Create a slider for V10 with a title of Percent. Label the left end of the slider Negative, the middle Neutral, and the right end Positive.

Example: Slider ( ,Lowest, ,Highest) V10 - V12  
(Create three grouped sliders for V10, V11, and V12. There will be no title and no middle labeling. The left end of the slider will be labeled Lowest and the right end will be labeled Highest. (Note that commas are used to indicate label separation, even if a label is blank).

Additional Examples:

Suppose in your codebook, you have a variable to measure satisfaction on a 1 to 10 scale. The codebook is:

*Variable 7 Label: How satisfied are you with the program?*

*Variable 7 Format: N2*

*Variable 7 Value Labels:*

*1=Not at all Satisfied*

*2=*

*3=*

*4=*

*5=*

*6=*

*7=*

*8=*

*9=*

*10=Very Satisfied*

The default script would be:

```
Radio (2) V7  
BlankLine
```

You would modify the default script to:

```
Slider V7  
BlankLine
```

The slider would be created and the endpoints would be labeled Not at all Satisfied and Very Satisfied. The slider would not have a title and the middle point of the slider would not be labeled.

The following would add a title to the slider:

```
Slider (Level of Satisfaction) V7  
BlankLine
```

In another example, suppose in your codebook, you have two variables to measure respondents' pessimism/optimism for their country and their company. You plan to use a scale of -10 to +10. The codebook would be:

```
Variable 1 Label: The country where you live.  
Variable 1 Format: N3  
Variable 1 Valid Codes: -10 - +10  
Variable 2 Label: The company where you work.  
Variable 2 Format: N3  
Variable 2 Valid Codes: -10 - +10
```

The default script would be:

```
TextBox V1  
BlankLine  
TextBox V2  
BlankLine
```

You would modify the default script to:

```
Question=Please indicate your level of pessimism or  
optimism by moving the slider handle with your mouse.  
Slider (Pessimism/Optimism Rating, Very Pessimistic,  
Neutral, Very Optimistic) V1 - V2  
BlankLine
```

Note: The slider object uses active scripting. Depending on your browser security settings, you may be asked for permission to run the script when previewing the HTML page on your local computer. Permission is not required to run the script after it has been uploaded to a Web hosting server.

## CheckBox for Multiple Response Variables

### **Syntax: *CheckBox Vx - Vy -or- CheckBox (2) Vx - Vy***

The CheckBox is used for multiple response. When creating Internet surveys with multiple response variables, there must be the same number of variables as there are value labels. Furthermore, all the value labels must be specified in each of the variables. In the following example, there would be 4 identical variables, each with 4 value labels.

The first form of the command will create all the CheckBoxes in a single column. The second form of the command will create two columns of CheckBoxes. In creating the default script, if there are six or fewer value labels, the default will be one column. If there are 7-12 value labels, the default will be two columns. The CheckBox is the only input method for multiple response.

4. Where did you hear about StatPac? (Check all that apply)
<input type="checkbox"/> Internet
<input type="checkbox"/> Magazine or Newsletter
<input type="checkbox"/> Word of mouth
<input type="checkbox"/> Other

## Checkbox for Groups of Multiple Response Variables (horizontal matrix)

### **Syntax: *Checkbox Vx - Vy***

Sometimes you might have several multiple response variables that are better shown in a horizontal matrix. For example, suppose you have the following question:

*What role did you play in selecting each of the following: (check all that apply)*

The codebook looks like this:

```
V9a_1 Label: Magazine A
1=Determine need
2=Recommend
3=Approve
V9a_2 Label: Magazine A
1=Determine need
2=Recommend
3=Approve
V9a_3 Label: Magazine A
1=Determine need
2=Recommend
3=Approve
```

*V9b\_1 Label: Magazine B*

*1=Determine need*

*2=Recommend*

*3=Approve*

*V9b\_2 Label: Magazine B*

*1=Determine need*

*2=Recommend*

*3=Approve*

*V9b\_3 Label: Magazine B*

*1=Determine need*

*2=Recommend*

*3=Approve*

*V9c\_1 Label: Other Magazine*

*1=Determine need*

*2=Recommend*

*3=Approve*

*V9c\_2 Label: Other Magazine*

*1=Determine need*

*2=Recommend*

*3=Approve*

*V9c\_3 Label: Other Magazine*

*1=Determine need*

*2=Recommend*

*3=Approve*

*V9c\_Other Label: Other*

The default script would treat each of the three groups of variables individually, and would look like this:

Example:

CheckBox Q9a\_1 - Q9a\_3

BlankLine

CheckBox Q9b\_1 - Q9b\_3

BlankLine

CheckBox Q9c\_1 - Q9c\_3

To switch to a matrix format, change the script so that all variables are in the same checkbox command.

Question=What role did you play in selecting each of the following:  
(check all that apply)

CheckBox Q9a\_1 - Q9c\_3

What role did you play in selecting each of the following: (check all that apply)			
	Determine need	Recommend	Approve
Magazine A	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Magazine B	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Magazine C	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

## Validity Checks for CheckBoxes

The `AtLeast` command is used when you want to require the respondent to check at least x CheckBoxes. The `AtLeast` command may be inserted anywhere in the script. The format of the command is: `AtLeast (minimum number of checks) Vx – Vy`. The minimum number of checks is enclosed in parentheses, and Vx and Vy should be the same as the `CheckBox` command. An example would be:

Question=Please select at least two of the following items.  
 CheckBox V20-V30  
 AtLeast (2) V20-V30

The `UpTo` command is used when you want to limit the number of CheckBoxes a respondent can check. The `UpTo` command may be inserted anywhere in the script. The format of the command is: `UpTo (maximum number of checks) Vx – Vy`. The maximum number of checks is enclosed in parentheses, and Vx and Vy should be the same as the `CheckBox` command. An example would be:

Question=Please select up to three items.  
 CheckBox V20-V30  
 UpTo (3) V20-V30

The `Exactly` command is used when you want to require the respondent to check exactly x CheckBoxes. The `Exactly` command may be inserted anywhere in the script. The format of the command is: `Exactly (number of checks) Vx – Vy`. The number of checks is enclosed in parentheses, and Vx and Vy should be the same as the `CheckBox` command. An example would be:

Question=From the following list, please select your three favorite items.  
 CheckBox V20-V30  
 Exactly (3) V20-V30

You might want to allow respondents to leave all the CheckBoxes unchecked. However, if they check any boxes they must check the number specified in the `AtLeast` or `Exactly` commands. The `Missing Allowed` field in the codebook controls this feature. When `Missing Allowed` is checked in the codebook for the first variable in the series (i.e., Vx), then the respondent can leave all the items unchecked. When

Missing Allowed is not checked, they will be required to check at least the number of CheckBoxes specified in the AtLeast command or exactly the number of CheckBoxes specified in the Exactly command.

## ListBox

### **Syntax: *ListBox Vx***

The ListBox may be used as an alternative to radio buttons. Functionally, the ListBox is identical to the radio buttons.



4. Where did you first hear about StatPac?

Internet  
Magazine or Newsletter  
Word of mouth  
Other

---

## Uploading and Downloading Files from the Server

### Auto Transfer

#### ***Uploading a Survey***

The easiest way to upload files to the Internet is to use the Auto Transfer feature. All necessary files will be uploaded automatically. Changes to the Perl scripts are made automatically.

Run Auto Transfer from the script window by clicking the Server button. Run Auto Transfer at other times by loading your codebook and then selecting Server > Auto Transfer.

If WebFolderURL (a Primary setting in the script) is set to your own server, Auto Transfer will be to and from that server. If WebFolderURL specifies the StatPac server, then Auto Transfer will be to and from your current folder on the StatPac server.

If you change private folders on the StatPac server, Auto Transfer will automatically adjust your HTML to reflect the new folder name. You do not have to regenerate the HTML or make other changes to your script.

#### ***Downloading Responses***

When you download responses using Auto Transfer they will be imported into StatPac. StatPac assumes that you are always downloading the entire data set from the beginning of the survey. Each time you download, StatPac will overwrite the existing data file with the newly downloaded data. Thus, there are only two situations where you should delete a response file from the server: 1) to delete the test data file, and 2) to delete the real data after you have finished the survey analysis.



You can edit the StatPac.ini file so that StatPac will prompt you on whether to overwrite or append to the existing data file. Set DataQuery = 1 to use the prompt.

## FTP

FTP (file transfer protocol) is the method used to transfer files to and from a remote Web server. Auto Transfer also uses FTP (it just happens "behind the scenes").

In order to use FTP, you need to know the locations of the wwwroot folder and the CGI folder on your server.

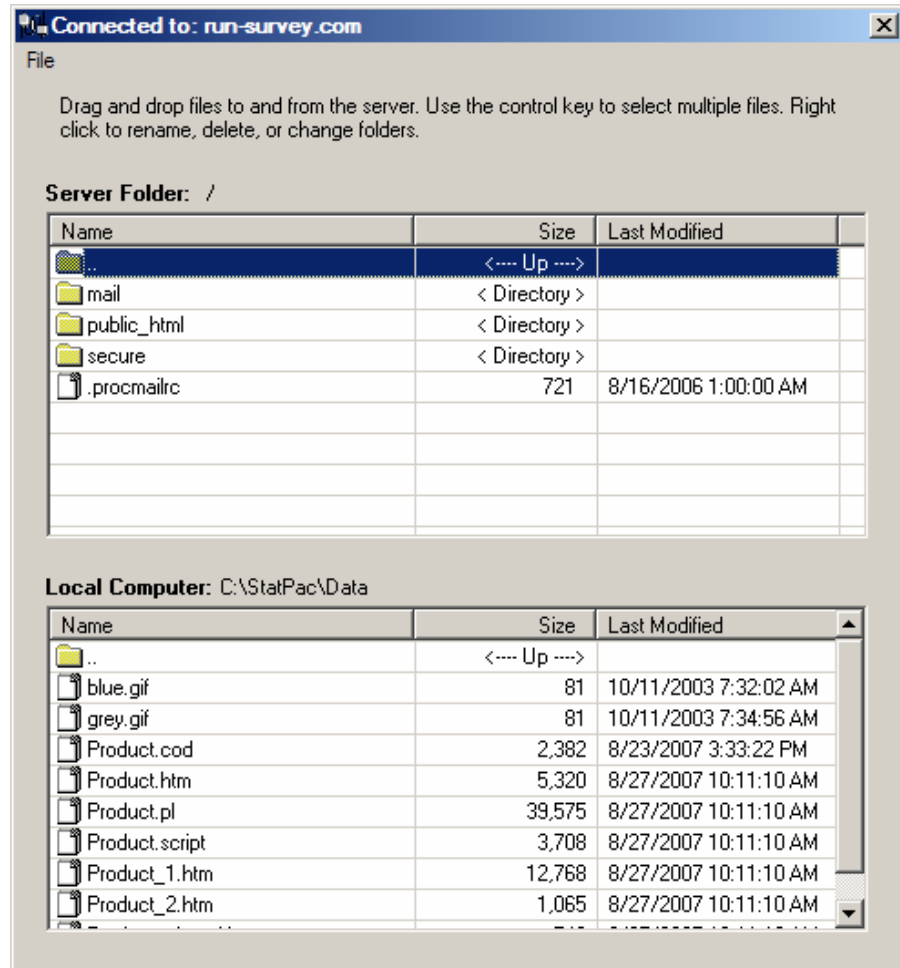
The wwwroot folder is where your Web site pages are located. On a Unix/Linux server this is probably public\_html. On a Windows NT or IIS server, it is probably wwwroot. When you login via FTP, you might already be in the wwwroot folder, or you might have to change to another subfolder (e.g., public\_html).

The CGI folder is almost always immediately below the wwwroot folder and it is usually called cgi-bin or cgi.

To run StatPac's FTP program, select Server > FTP > *server*

Two panes will be displayed. The top pane is your server and the bottom pane is your local computer. You can drag and drop files from one pane to the other. Highlight one or more files in one pane and drag those to the other pane. The current folder is shown in the text on top of the pane. That is the folder where files will be dropped. To drop to a displayed folder, first double click on that folder to make it the current folder.

You can also right click on a file in either pane for additional options. Double click on a folder to change to that folder.



When using the StatPac server, you will only have access to two server folders, 1) the folder where your surveys reside, and 2) the folder where your response files are written. Select View to change folders.

Demo users all share the folder name of *guest* and therefore will not have FTP access to the StatPac server. When using your own server, you will have complete FTP navigation.

### Uploading a Survey

The WebFolderURL setting specifies the folder where the survey will reside. If it is just a URL without a folder name, then the survey should be uploaded to the wwwroot folder.

**WebFolderURL=http://yourserver.com**

However, if it also specifies a folder name, then the HTML files should be uploaded to that folder.

**WebFolderURL=http://yourserver.com/survey**

While Auto Transfer creates the folder if it doesn't exist, you may have to manually create it if using FTP. In the server pane (top), change to the wwwroot folder (if necessary). Then right click on the pane and select New Folder.

All of the following files will be found in your current project folder.

The following files need to be uploaded to the survey folder.

SurveyName.htm (Loader)  
SurveyName\_1.htm (1st page)  
SurveyName\_2.htm (2nd page)  
SurveyName\_3.htm (3rd page)  
SurveyName\_4.htm (Thank-you page)

Additionally, the HTMLName command is used to name several other files.

SurveyName\_closed.htm (Survey is closed page)  
SurveyName\_cookie\_cutter.htm (Delete the cookie)  
SurveyName\_popup\_1.htm (1st popup window)  
SurveyName\_popup\_2.htm (2nd popup window)  
closepopup.htm (internal file to close popup windows)  
SurveyName\_help\_1.htm (1st help window)  
SurveyName\_help\_2.htm (2nd help window)  
SurveyName\_start.htm (Loader page for password protected surveys)  
Any graphics used (including blue.gif and grey.gif)

The following file needs to be uploaded to the cgi-bin folder.

SurveyName.pl (Processes data when respondent clicks submit)

After uploading the .pl file, right click on it and set the permissions to 755.

If you have a password protected survey, you also must upload two additional files to the cgi-bin folder:

password.pl (Controls the login)  
SurveyName.text (Data base containing one or more passwords)

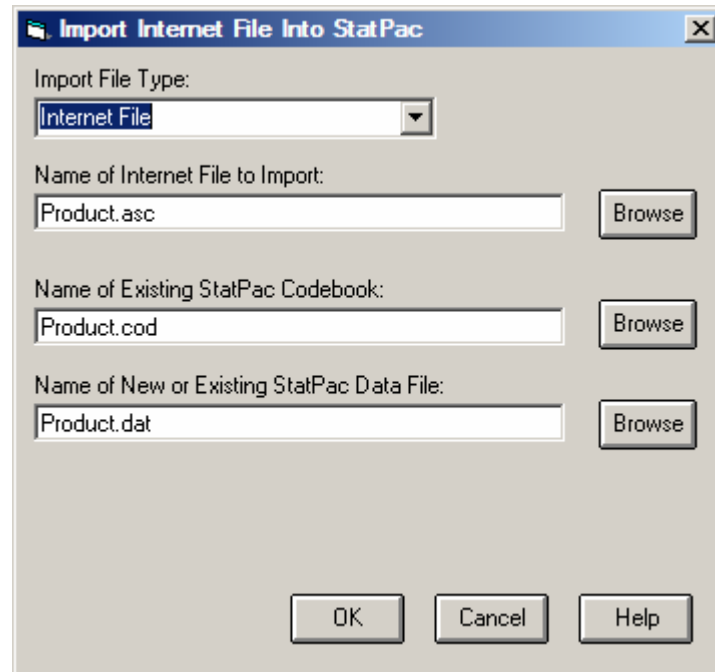
After uploading password.pl, right click on it and set the permissions to 755.

## ***Downloading Responses***

Responses will always be saved in the cgi-bin folder and have a .asc extension.

In the server pane (top), navigate to the cgi-bin folder, drag the SurveyName.asc file to your current project folder.

While Auto Transfer automatically imports the data, when you use FTP, you have to manually import the .asc file to create the data file. Close the FTP window and select Data > Import.



---

# Summary of the Most Common Script Commands

This is a summary of the most commonly use commands in the Survey Creation section of the script.

## Commands to display text

Heading=Text	Displays very large text
Title=Text	Displays large text
Text=Text	Displays normal text
Footnote=Text	Sets the footnote text
Instructions=Text	Displays text using instructions color scheme
Question=Text	Displays text using question color scheme
HTML=Any valid HTML	Displays text or HTML elements

## Commands for spacing and pagination

BlankLine	Inserts blank line
NewPage	Begins a new survey page

## Commands to insert images and links

Image=Filename(s)	Insert one or more images
LinkURL=URL	Sets the URL for the next hyperlink
LinkText=Text	Sets the text for the next link on the survey
Link	Inserts the hyperlink in the survey

## Commands for help and popup windows

HelpLinkText=Text	Sets the text for the help link on the survey
HelpText=Text	Sets the text that will appear in the popup help window
HelpFileName=Filename	Sets the name of an existing popup help window file
HelpWindow	Inserts the help window hyperlink into the survey
PopupStart	Begin a popup window
PopupEnd	End a popup window

## Commands to create objects

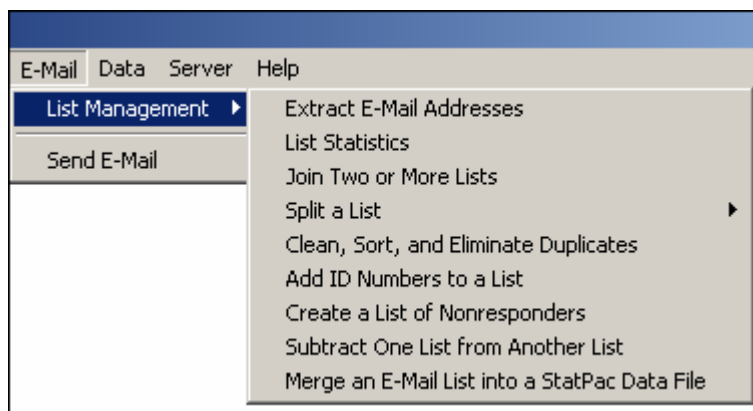
Radio Vx	Inserts radio buttons for variable x
Radio Vx-Vy	Inserts horizontal radio buttons for variables x-y (matrix format)
DropDown Vx	Inserts a dropdown box for variable x
TextBox Vx	Inserts a single or multiple line text box for variable x (depending on the field width of the variable)
TextBox Vx-Vy	Inserts text boxes for variables x-y into a single frame
Slider Vx-Vy	Inserts sliders for variables x-y into a single frame
ListBox Vx	Inserts a list box for variable x

# Email List Management

---

## Overview

StatPac contains a complete e-mail list management system. Select, Email, List Management to access these features.



---

## Format of an Email Address File

An e-mail list is simply an ASCII text file of e-mail addresses. The file may optionally contain other information. Email address lists may use a .lst, .txt, or .csv extension for the file name. When StatPac creates or writes an e-mail list, it will always use a .lst extension. These are tab delimited ASCII text files. If necessary for compatibility with other software packages, .lst files may be renamed to .txt.

The file should contain one record (line) for each person that will be sent an e-mail. In its simplest form, this is just a file of e-mail addresses (one per line). An address file with three names might look like this:

david@statpac.com

john\_smith@mindspring.com  
fred289@aol.com

Optionally, the text file can also contain a unique ID number or identifier. If included, it should be separated from the email address by a comma or a tab. In other words, it is a comma or tab delimited file with two variables. The first variable is the email address and the second variable is a unique ID. The unique ID can contain numbers and letters, but not special characters like question marks, pound symbols, or spaces. The ID number will always be the second variable in an e-mail address line.

david@statpac.com,912783  
john\_smith@mindspring.com,7576  
fred289@aol.com,4568063

Additional information may also be included in the e-mail address file. Each additional field is separated from the other variables by a comma or tab. All records in the e-mail address file must contain the same number of variables. One reason you might have additional variables in an e-mail address file is to be able to customize the e-mails. For example, instead of “Hello Customer”, you could begin your e-mail with “Hello David”. Another reason you might have additional information is to be able to merge that data with the completed surveys. That is, you can join data from an existing data base with the answers to a survey.

Additional information in an e-mail address file always begins with the third variable. The first variable is the email address, the second variable is the ID number, and the additional information begins is the third variable. For example, this e-mail address file contains the first and last names, and these could be used in the body of the e-mail.

david@statpac.com,912783,David,Walonic  
john\_smith@mindspring.com,7576,John,Smith  
fred289@aol.com,4568063,Fred,Jones

In all of the above examples, the first record in the e-mail address file contained actual data. Many other software programs write a header row when creating an delimited text file (i.e., an e-mail address file). That is, the first record in the e-mail address file contains variable names instead of data. An example would be:

Email,ID,First,Last  
david@statpac.com,912783,David,Walonic  
john\_smith@mindspring.com,7576,John,Smith  
fred289@aol.com,4568063,Fred,Jones

StatPac can be set to use or not use a header row for all e-mail address lists. Edit StatPac.ini and set EmailListHeaderRow = 1 to use a header row or EmailListHeaderRow = 0 to not use a header row.

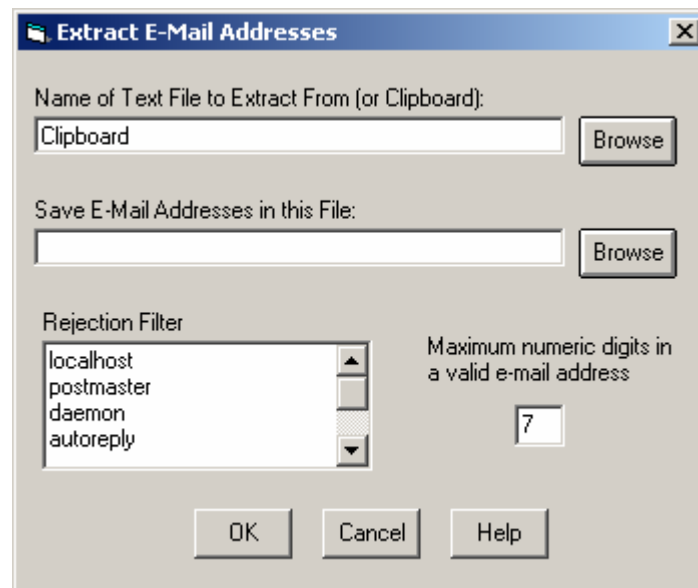
---

## Extract Email Addresses

It is often desirable to be able to extract e-mail addresses from a file or from the clipboard. This is especially helpful if you are building an e-mail list. It can be used to capture bounced e-mail addresses that you want to eliminate from an address file. The file you want to extract e-mail addresses from can be in any text format. StatPac will find and extract the addresses and write them to an e-mail address file (.lst extension). The extract program will not extract e-mail addresses from compressed data base formats such as Access. You should export the data to a text file before attempting to extract the e-mail addresses.

There are two problems you might encounter when extracting e-mail addresses. The first is that you might extract e-mail addresses that are generated by a server. For example, postmaster@somedomain.com is probably not an address you'd want to add to an e-mail address file. Your own domain is probably also an address you want to exclude. The rejection filter may be used to omit those addresses/domains that you do not want to be added to an e-mail address file even when they exist in the document you are extracting from. The rejection filter should contain one word per line. If that word is part of an e-mail address, it will not be captured by the program. Rejection filter text is not case sensitive.

The other problem you might encounter is that many servers issue a long numerical e-mail address as part of the header for a bounced or rejected e-mail. These might look like this: 1736846748836494774@somedomain.com. You can set the maximum consecutive digits parameter to filter out long numerical e-mail addresses generated by a server.

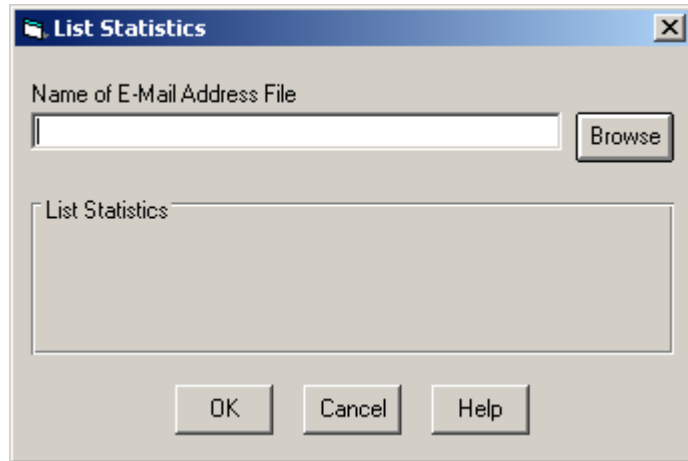


---

## List Statistics

The List Statistics program can be used to count the number of e-mail addresses in a file and tell when the last update was made to that file.



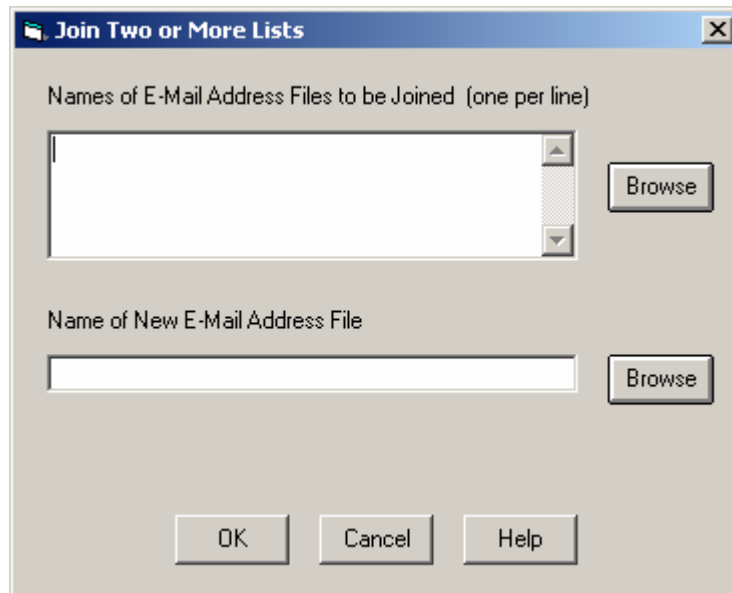


---

## Join Two or More Lists

There will be many situations where you have gathered e-mail addresses from multiple sources and stored them in separate files. Use this program to combine lists from multiple files and create a new file consisting of addresses from all the lists. Select the files to be joined using the Browse Button, or type them one per line. Specify a new e-mail address file that will contain the addresses from all the files.

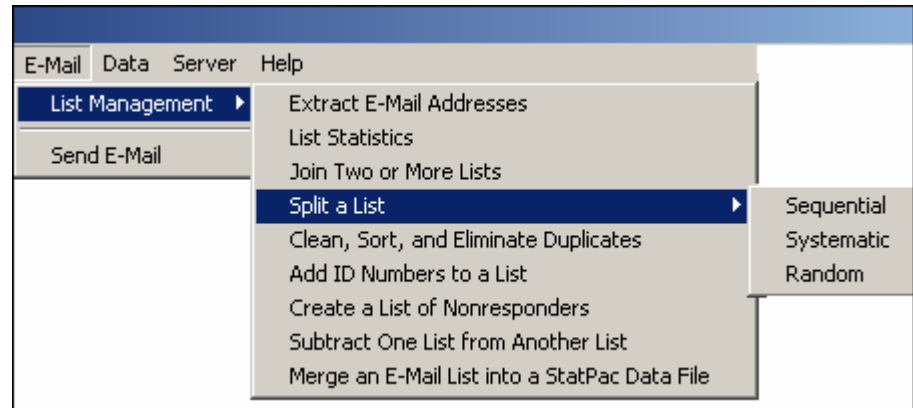
This program does not check for the existence of duplicate e-mail addresses. It only combines the files into one big list. Therefore, after combining multiple e-mail files, it is advisable to run the e-mail management program to clean, sort, and eliminate duplicates.



## Split a List

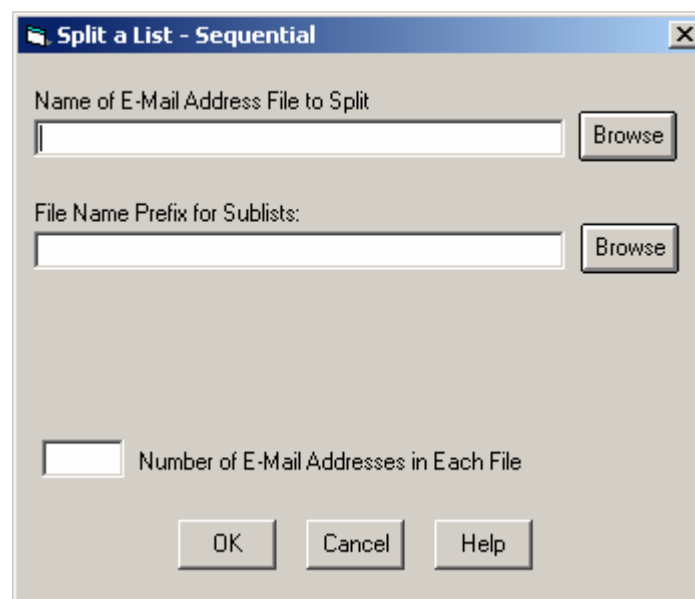
There are many situations where you might want to split a list: 1) You have a very large list and you want to e-mail to only a specific number of respondents 2). You have a very large list and you want to split it up to make it more manageable, 3) You want to randomly select winners for a drawing, etc.

StatPac can split a list using three different methods: sequential, systematic and random. Which method you use depends upon your application.



When the sequential method is selected, a new file will be created and e-mail addresses will be written to the new file until a certain number of addresses have been written. Then another new file will be created and the next  $x$  addresses will be written to that file. Each of the new e-mail address files will contain the same number of records as the previous file(s), except the last one that is written.

The File Name Prefix for Sublists is the beginning of the file name for the new smaller lists. The actual file names for the sublists will end with \_1, \_2, \_3, etc.



Systematic selection is often called Nth Name Selection. Every Nth name will be selected from an e-mail list file and written to a new e-mail list file. Optionally, you can also write the non-selected addresses to a different e-mail address file. Systematic selection is frequently used to draw a sample for a survey from a larger list. It is considered to be as good as random selection provided that the order of the e-mail list is not related to the focus of the study.

The dialog box is titled "Split a List - Systematic". It contains three text input fields, each followed by a "Browse" button. The first field is labeled "Name of E-Mail Address File to Split". The second field is labeled "Name of File for E-Mail Addresses that are Selected:". The third field is labeled "Name of File for E-Mail Addresses that are Not Selected:" with "(Optional)" below it. Below these fields is a text input field labeled "Value of N (Select Every Nth E-Mail Address)". At the bottom are three buttons: "OK", "Cancel", and "Help".

The third method of splitting a list is the random selection method. Records will be randomly selected from the e-mail list. You only need to specify the number of random records that will be drawn from the e-mail address file. Optionally, you may also specify a file name for the records that are not randomly selected.

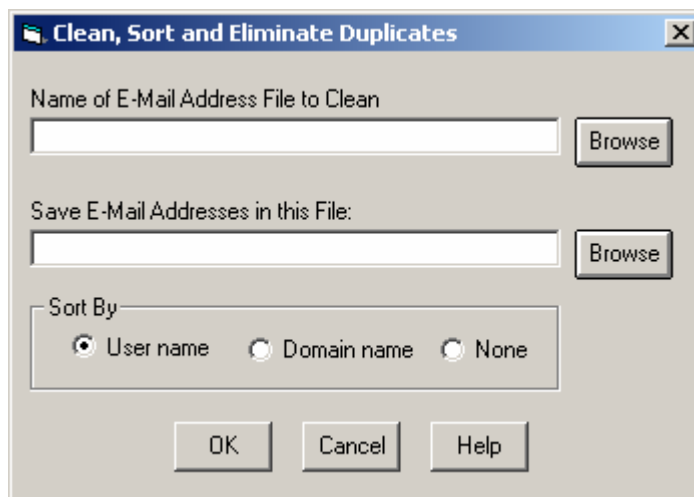
It should be noted that this method involves true random selection. Thus, running the program twice on the same list will not select the same e-mail addresses.

The dialog box is titled "Split a List - Random". It contains three text input fields, each followed by a "Browse" button. The first field is labeled "Name of E-Mail Address File to Split". The second field is labeled "Name of File for E-Mail Addresses that are Selected:". The third field is labeled "Name of File for E-Mail Addresses that are Not Selected:" with "(Optional)" below it. Below these fields is a text input field labeled "Number of E-Mail Address to Select". At the bottom are three buttons: "OK", "Cancel", and "Help".

---

## Clean, Sort, and Eliminate Duplicates

Many e-mail lists are dynamic, in that you are continually updating, adding, and deleting e-mail addresses from the list. The Clean, Sort, and Eliminate Duplicates program can be used to maintain a clean list. Invalid and duplicate e-mail addresses will be eliminated.



---

## Add ID Numbers to a List

ID numbers are used to keep track of those who respond to your survey. You can conduct web surveys without ID numbers, but there will be no way to determine who responded and who didn't. If you are planning on sending e-mailing follow-up invitations to those who did not respond to the first invitation, then ID numbers are essential. ID numbers are used to link the names in an e-mail address file to the survey respondents.

If you already have an ID number in an existing data base, it can be used provided that it does not contain any special characters, like question marks, pound symbols, or spaces. It must be the second variable in the e-mail address file (which is tab or comma delimited ASCII text).

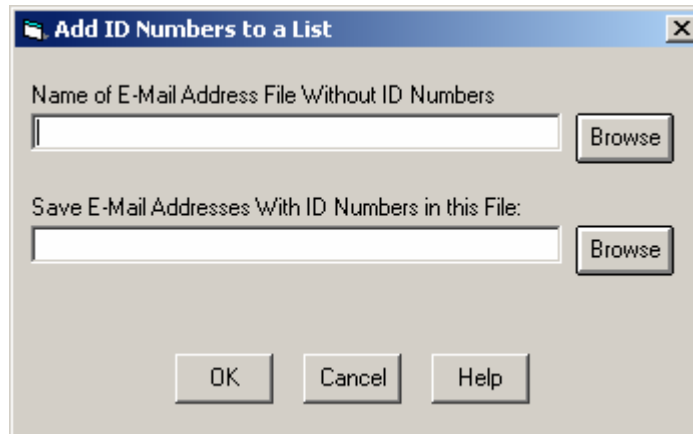
The Add ID Numbers to a List program may be used to create random ID numbers and add them to an existing e-mail address list. The ID numbers will be inserted as the second variable in the list. If there is already more than one variable in the list, the ID number will be inserted as variable two and all the other variables will be moved to the right. For example, suppose you have this e-mail list consisting of three variables (e-mail address, company, and city).

david@statpac.com, StatPac Inc., Minneapolis

john\_smith@mindspring.com, Acme Widgets Co., Chicago

After adding ID numbers, the list would appear like this:

david@statpac.com, 21769831, StatPac Inc., Minneapolis  
john\_smith@mindspring.com, 89457912, Acme Widgets Co., Chicago



---

## Create a List of Nonresponders

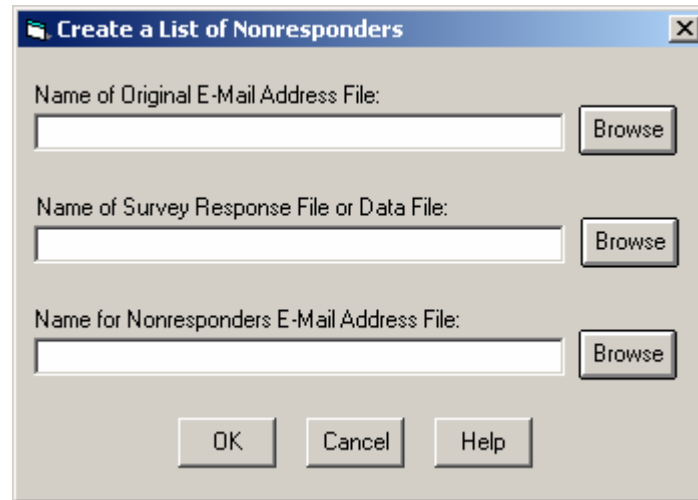
In a typical web survey, you might send out hundreds (or thousands) of e-mail invitations to take the survey. Within a few days, you will receive about 90% of the total response that you'll get. After a week has passed, you can substantially boost the response rate by sending a reminder e-mail to those who have not yet responded. The purpose of this program is to create an e-mail list consisting of just those people who have not yet responded to the survey.

The process is straight forward, although there are several steps involved.

- 1) Add the *RespondentID* variable to the codebook. When you create an Internet survey, StatPac will ask if you want to add the special variables to the codebook if they have not already been added. Generally, you would answer yes.
- 2) Run the e-mail management program to add ID numbers to the e-mail address file.
- 3) Send e-mail invitations using StatPac's bulk e-mail program.
- 4) Approximately one week later, download and import the Internet response file (Auto Transfer).
- 5) Run this program and create an e-mail list of nonresponders. This list will be identical to the original e-mail list except it will only contain the e-mail addresses of those who have not yet responded to the survey.
- 6) Send a reminder e-mail to the nonresponders using StatPac's bulk e-mail program. The only two parameters in the bulk e-mail program you will need to change are the name of the e-mail address file (which will be the nonresponders file instead of the original file) and the file containing the body text (which will presumably be different than the original invitation).

7) After another week, re-download the Internet response file. When you download an Internet response file from your server, it is not deleted from your server. Thus, when you re-download the Internet response file, you should overwrite the existing data file rather than appending to it.

8) Analyze the data.

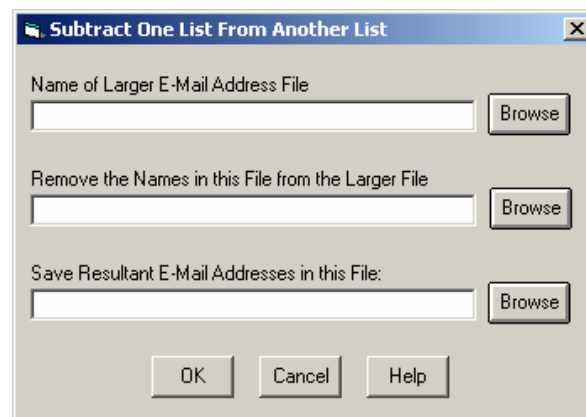


---

## Subtract One List From Another List

It is very common for e-mail addresses to be wrong. People change accounts and e-mail addresses frequently. Typing errors in e-mail addresses are abundant. It would not be unusual for 10%-20% of the addresses in an e-mail address file to be invalid.

When you send bulk e-mail to the names in an e-mail address file, invalid addresses will bounce back to you. These bounced e-mails can be captured with the e-mail management program to extract e-mail addresses. After you have an e-mail list of the bounced e-mails, you will probably want to remove them from the e-mail address file that was used to send the invitations. This is especially true if you want to use the e-mail list again in the future. You can subtract (i.e., remove) the bounced e-mail addresses from the e-mail file using this program.



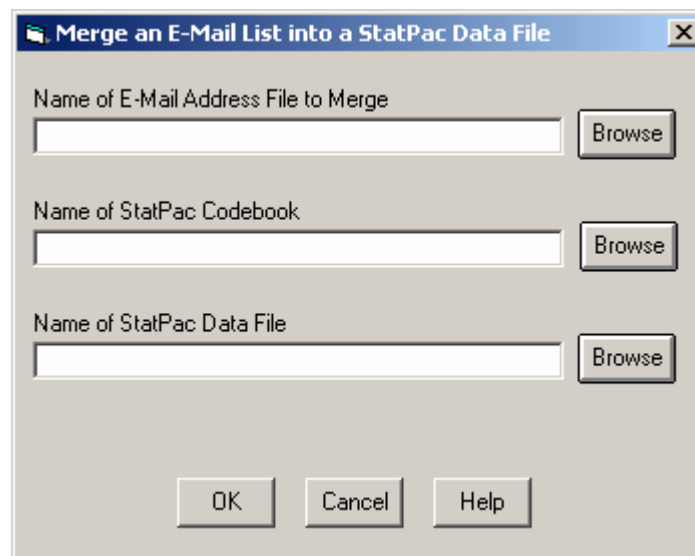
---

## Merge an Email List into a StatPac Data File

When you receive responses to an Internet survey, the file will not contain the e-mail address of the respondent. In fact, there is no way to capture the e-mail addresses of people taking the survey unless the survey itself specifically asks for their e-mail address as one of the variables.

However, if you used Respondent ID numbers, you'll be able to match respondent's with their entry in the e-mail address file. The e-mail address file will contain the respondent's e-mail address, ID number, and possibly other data base information.

The e-mail list management program will merge the e-mail list information into the StatPac data file. You should wait until the survey is closed before using this program. In other words, do not run this program if you are expecting additional response.



---

## Send Email Invitations

Email is a popular way of inviting potential respondents the opportunity of participating in an Internet survey. The Send Email Invitations program will allow you to send a large number of customized e-mails at high speed.

Before beginning, you need two files. The first is a list of e-mail addresses. The second is an ASCII text file containing the body text of the e-mail. Usually, you would create this file with Notepad or MSWord.

### Using an ID Number to Track Responses

One important feature of the e-mail program is that it lets you serialize the e-mails with a unique ID number that can be used to track respondents.

In the body of the e-mail, there will be a link to the survey on the web site. This link will contain the respondent's unique ID number, so when they click on the link, it will take them to the survey web page (and StatPac will know the ID number of the

person who is responding). Moreover, their ID number will be stored with their responses on your server. The ID number may contain alpha characters as well as numbers. This feature can be used to match information from an existing data base with respondents' answers. In the following example of a link with an ID number, notice the URL has a ?id=1856 suffix and the ID number is 1856.

<http://www.statpac.com/online-surveys/multipage-survey.htm?id=1856>

## Email Address File

It is VERY important that you create a short test file of one to three e-mail addressees. All the e-mail addresses in the test file should be your own e-mail address. This will allow you to send test e-mails to yourself to check their appearance before you send to the real respondents. Name this file something like "TestList.lst" so you will immediately know what it is by looking at its name. You can also use the Test Mode by typing your e-mail address into the text box so that instead of sending to the specified e-mail list, only one e-mail will be sent to you. Be sure to completely erase your e-mail address from the text box when you are ready to send to the entire list.

## Body Text File

The body text file is the actual e-mail message you will be sending to potential respondents. It may be plain text or HTML. You can use Notepad or MSWord to create a plain text body file, or MSFrontPage of MSWord to create an HTML email. For a plain text email, StatPac can include attachments to the email. If it is an HTML email, StatPac will automatically include the graphics in the body of the email. When using an HTML body file with graphics, the "src=" tag in the HTML source should point to the image on your local computer (not the internet).

Variable substitution can be used in the body text file to include the URL of the survey. If the address file is a StatPac data file, then you can also use variable substitution for any variable in the codebook. Substitution is specified using (\* and \*) to begin and end the variable name.

Here is an example of a body text file that uses three substitution variables. Assume variable 3 in the e-mail address file is "First\_Name" and variable 4 in the e-mail address file is "Last\_Name". In this example, variable substitution is being used to customize the greeting. It also uses (\*url\*) to specify where the link should be placed in the e-mail. The (\*3\*) and (\*4\*) can be specified in the body text because they are variables in the e-mail address file.

Hello (\*3\*) (\*4\*),

We are doing a survey and would like your participation.

Please click here to take the survey

(\*url\*)

Thank you.



Regards,

David Walonick  
StatPac Inc.  
<http://www.statpac.com>  
(715) 442-2261

If the email is HTML then the link to the URL is still (\*url\*). For example, the source in the HTML might look like this:

```
<a href="(*url*)">Please click here to take the survey</a>
```

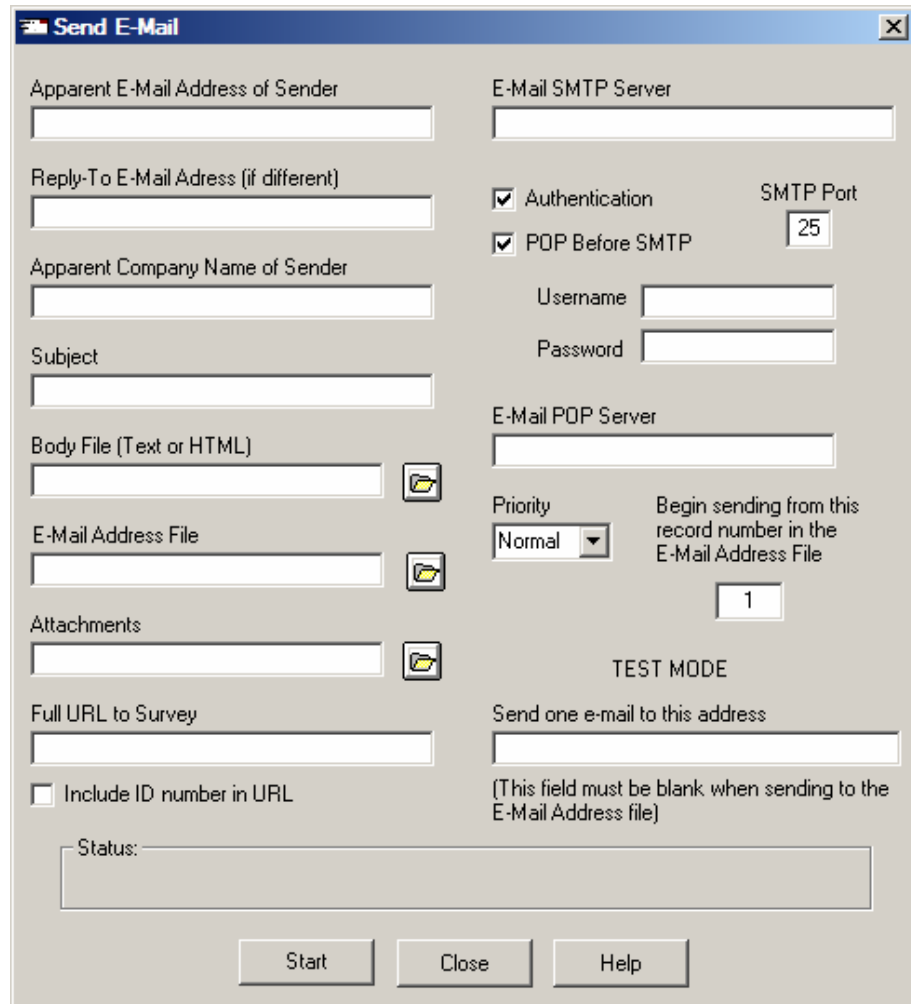
If the e-mail address file contains a header row of variable names, then the names in the header row can be used for variable substitution instead of the variable number. For example, the greeting could be written as:

Hello (\*First\_Name\*) (\*Last\_Name\*),

When creating a plain text body file, we recommend limiting all lines to a maximum of 60 characters and inserting a hard return at the end of each line. (This means to press [Enter] after each 60-character line when you are typing the body text). While not necessary, it will produce a more uniform appearance in the variety of e-mail readers that potential respondents might be using.

## **Sending Email**

After you have created the e-mail address and body text files, you'll be ready to test your work. Select Email, Send Email to begin show the bulk e-mail sending screen.



The image shows a Windows-style dialog box titled "Send E-Mail". It contains several input fields and checkboxes for configuring an email send operation. The fields are arranged in two columns. The left column includes fields for "Apparent E-Mail Address of Sender", "Reply-To E-Mail Address (if different)", "Apparent Company Name of Sender", "Subject", "Body File (Text or HTML)", "E-Mail Address File", "Attachments", "Full URL to Survey", and a "Status:" label with a text area. The right column includes fields for "E-Mail SMTP Server", "SMTP Port" (set to 25), "Username", "Password", "E-Mail POP Server", "Priority" (set to Normal), and a "TEST MODE" section with a "Send one e-mail to this address" field. There are also checkboxes for "Authentication" and "POP Before SMTP", and a "Begin sending from this record number in the E-Mail Address File" field (set to 1). At the bottom are "Start", "Close", and "Help" buttons.

Apparent E-Mail Address of Sender	E-Mail SMTP Server
Reply-To E-Mail Address (if different)	<input checked="" type="checkbox"/> Authentication
Apparent Company Name of Sender	<input checked="" type="checkbox"/> POP Before SMTP
Subject	SMTP Port: 25
Body File (Text or HTML)	Username
E-Mail Address File	Password
Attachments	E-Mail POP Server
Full URL to Survey	Priority: Normal
<input type="checkbox"/> Include ID number in URL	Begin sending from this record number in the E-Mail Address File: 1
Status:	TEST MODE
	Send one e-mail to this address
	(This field must be blank when sending to the E-Mail Address file)

Start Close Help

The *Apparent Email Address of Sender* should contain the e-mail address that you want to appear as the sender of the e-mails. Similarly, the *Apparent Company Name of Sender* should be set to the name you want to appear as the sender. *Subject* is what will appear on the subject line of the e-mails.

If you want the *Reply To Email Address* to be different than the *Apparent Email Address of Sender*, then enter a different e-mail address. If it is the same, then leave this field blank. The *Reply To Email Address* is the address that the respondent will see if they click the reply button in their e-mail program.

Use the browse button to select the *Body Text File*. This can be a plain text file or an HTML file.

Use the browse button to select your "TestList.txt" file for the *Email Address File*. Always begin with your test list until you are satisfied with the appearance of the e-mails. Then change *Email Address File* to your actual list of e-mail addresses. It is generally a good idea to limit the size of your *Email Address Files* to less than five thousand addresses, as some servers stop responding to extended SMTP sending.

If sending a plain text body file, you can also specify *Attachments*. Multiple attachments may be specified by clicking the Attachments browse button multiple times or by holding down the control key to select multiple attachments.

*Full URL to Survey* is the URL of the survey on your web site. It should be the fully qualified path to your survey, beginning with `http://`. An example would be: `http://www.statpac.com/online-surveys/multipage-survey.htm`. If you are using password protection, the link might be `http://www.statpac.com/cgi-bin/multipage-survey.pl`. Do not inadvertently end the URL with a period. The URL is case sensitive, so type it exactly as it is on your web site. Do not add a query string to the URL. That is, do not include a question mark in the URL. The query string is reserved for the respondent ID number and will be added automatically by the software during the send.

If your e-mail list has ID numbers (as the second variable), check the *Use ID Numbers in URL* box. The ID number will become the query string during the sending of the e-mails. If you do not check the box, no ID numbers will be used in the e-mails and a random ID number will be assigned when a respondent clicks on the link.

*Email SMTP Server* is the mail server you will be using. It will be something like `mail.yourdomain.com`. Your ISP will be able to tell you what to specify here. Alternatively, you can check the SMTP settings in your own e-mail program and set StatPac to the same thing.

If your server requires authentication to send emails, and check *Authentication* box type your Username and Password. Some servers require “POP-before-SMTP” protocol. If the StatPac’s bulk email program won’t send, this is the most likely reason. To fix this problem, check *POP Before SMTP* and type the name of your POP3 mail server (often the same as your SMTP server). When “POP-before-SMTP” protocol is used, ISPs usually limit sessions to about half hour, so email address list sizes should be limited to what can be sent in that time.

*SMTP Port* is the port used by your SMTP server. The standard is port 25, but occasionally ISPs will use a different port number. Do not change this unless your ISP requires the use of a different port.

You must be logged onto your server before you begin sending the e-mails. Click *Start* to begin sending the e-mails.

When you are satisfied with the e-mail appearance (by sending several to yourself), change the *Email Address File* to your real file and click *Start* to begin sending the e-mails. The status window will show the progress.

The speed at which the e-mails are sent will depend on your e-mail SMTP server. StatPac can easily send three or more e-mails per second. This may be too fast for some servers and you may have to slow StatPac down a bit. To slow StatPac down, edit the *StatPac.ini* file (select *File / Open / System Defaults File*) and change *EmailDelay* from 0 to a delay value (in thousands of a second). For example, if you set *EmailDelay* = 100, then there will be a 1/10 of a second delay between each email.

While sending e-mails, you can temporarily pause by clicking the *Pause* button. If you click the *Resume* button, the e-mails will continue to be sent from the point where you paused. If you close the *Send Email Invitations* window, make note of the record number so you will be able to continue where you left off.

Some e-mails might fail to be sent because of either bad addresses or the server becomes temporarily busy. Failed e-mails will be saved in a new *Email Address File* with a “-FAILED” suffix as part of the file name. You may attempt to resend these by changing the *Email Address File* to the failed list.

Three words of caution are in order.

1. StatPac's Send Email Invitations program is a general purpose bulk e-mailer that can send thousands of e-mails per hour. If you use it to send SPAM to people, there is a good chance that someone will contact your ISP, and the most likely result is that your ISP will cancel your account. Please don't use this program to send SPAM.
2. Some ISP servers are set to automatically detect and refuse bulk sending of emails. They detect multiple emails being originated by the same SMTP server connection. Two StatPac settings can be used to minimize the possibility of being flagged as SPAM by your ISP. Both are in the StatPac.ini file (select File / Open / System Defaults File). If EmailDelay is set to a value higher than zero, a new SMTP connection will be established for each email. If EmailLimitPerConnection is set to a value higher than 0, the software will close and reopen the SMTP connect at that limit. You can set either or both values.
3. Some hosting services have a quota on the number of emails you can send in a day. They do this in order to prevent spammers from abusing their SMTP mail server. If your ISP has such a policy, they will usually increase your quota by a simple request stating the reason why you need a higher quota.

# Procedure Files

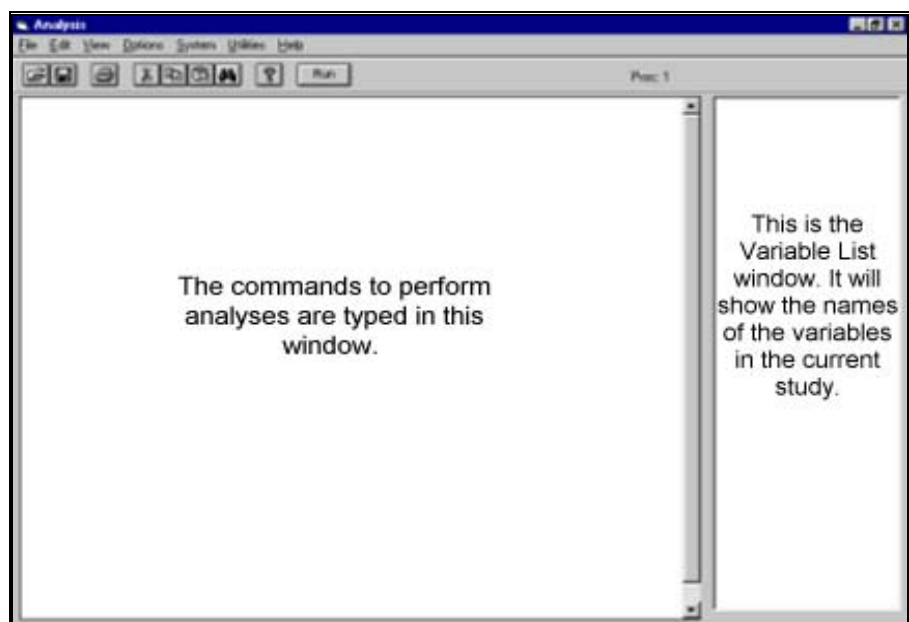
---

## Overview

A procedure in StatPac refers to a set of commands that perform one or more tasks. A procedure may specify a single analysis or several analyses of the same type. Procedures can also contain commands to perform transformations and write subfiles.

The commands to perform an analysis (or series of analyses), can be stored in a file called a "procedure file". This means that you can easily recall a previously executed procedure, and make changes to it without having to retype the commands. The procedure file is automatically stored on disk with the study name and a .PRO extension. You can also store procedure files using different names. Procedure files can be saved, loaded and merged with other procedure files.

Click on the Analysis Button to start the procedure file editor. The commands to run analyses are typed into the text window on the left of the screen. The Variable List window (on the right of the screen) will show the names of the variables in the current study.



---

## Mouse and Keyboard Functions

The procedure file editor is similar to any text editor, although it has numerous built in features to simplify editing procedure files.

### ***Mouse Functions***

Click	Move cursor to point of click
Shift+Click	Extend selection to the point of click.
Double-click	Selects the word that is clicked on (when no variables are selected in the Variable List window); otherwise, transfers selected variables from the Variable List window to the text.
Drag	Select text from point of button down to point where button is released.
Double-click and drag	Extend the selection from word to word.
Triple-click and drag	Extend the selection from row to row.

### ***Keyboard Functions***

HOME	Move cursor to the beginning of the line.
END	Move cursor to the end of the line.
(Left Arrow)	Move cursor one character to the left.
(Right Arrow)	Move cursor one character to the right.
(Up Arrow)	Move cursor one line up.
(Down Arrow)	Move cursor one line down.
CTRL+(Left Arrow)	Move cursor to the beginning of the current word.
CTRL+(Right Arrow)	Move cursor to the beginning of the next word.
CTRL+HOME	Move cursor to start of text.
CTRL+END	Move cursor to end of text.
CTRL+N	Move cursor to next procedure
CTRL+P	Move cursor to previous procedure
DEL	Delete selected text.
CTRL+X or SHIFT+DEL	Copy selected text to the Clipboard and delete the selection.
CTRL+C or CTRL+INS	Copy selected text to the clipboard.
CTRL+V or SHIFT+INS	Insert text from the clipboard.
CTRL+(Backspace)	Delete previous word.

---

## Designing Analyses

StatPac uses an easy programming language for designing procedures. It also has a large selection of automatic features to simplify the process.

For example, a simple procedure might be:

```
STUDY SURVEY
FREQUENCIES RACE
..
```

This procedure consists of a single task using a study called SURVEY (i.e., a codebook called SURVEY.COD and a data file called SURVEY.DAT). The procedure says to use the study called SURVEY, and perform a frequency analysis of the RACE variable. Notice that a procedure always ends with two dots (periods).

A more complex procedure would be:

```
STUDY SURVEY
FREQUENCIES RACE, INCOME, SEX
..
```

This procedure contains three tasks, each using the same study (and data file). If you execute this procedure, the program will first do a frequency analysis of RACE, then of INCOME, and finally of SEX. There is no limit to the number of tasks that can be specified in a single procedure.

A procedure file may also contain many different procedures. The only requirement is that the procedures be separated from each other by two dots. For example, the following commands specify three procedures:

```
STUDY SURVEY
FREQUENCIES RACE
..
CROSSTABS RACE BY AGE
..
FREQUENCIES AGE, INCOME, SEX, PREFERENCE
..
```

The first procedure contains one task, the second one task, and the third four tasks. These commands would actually run six analyses. Notice that the study name is only specified once (in the first procedure). Subsequent procedures will automatically use the same study and data file. Usually, the STUDY command is used only once.

The use of the STUDY keyword in the first procedure is mandatory since it defines the codebook and data file names for all the following procedures. However, the STUDY keyword may also be used in subsequent procedures. If the keyword STUDY is specified in another procedure, that procedure, and the procedures following it, will use the new codebook and data file.

The following commands contain two procedures, each having two tasks. The STUDY command is used in both procedures. This means that the first procedure will analyze data from one study (SURVEY1) and the second procedure will analyze data from another study (SURVEY2).

```
STUDY SURVEY1
FREQUENCIES AGE, PREFERENCE
..
STUDY SURVEY2
FREQUENCIES AGE, PREFERENCE
..
```

The STUDY keyword not only specifies the name of the codebook to be analyzed, but it also implicitly specifies the name of the data file. In most cases, the codebook and the data file name are the same (except for the extensions). A codebook called SURVEY would usually use a data file called SURVEY.DAT.

Sometimes, the codebook name and data file names will not be the same. For example, if the same study had been performed each year, you might have several data files with the same codebook name, but with different data file names. The DATA keyword may be used to analyze different data files (all using a common codebook name).

```
STUDY SURVEY
DATA DATA-97
FREQUENCIES AGE, ATTITUDE
..
DATA DATA-98
FREQUENCIES AGE, ATTITUDE
..
```

In the above example, each procedure uses the DATA keyword to specify what data file should be analyzed by that procedure. Both procedures will use the codebook called SURVEY.COD. Even though the STUDY command is only specified in the first procedure, subsequent procedures will use the same codebook name unless another STUDY keyword is used to change it. The first procedure will read data from a file called DATA-97.DAT and the second procedure will read from a data file called DATA-98.DAT.

Whenever you use the STUDY or DATA commands, the last specification will remain in effect until changed by another STUDY or DATA command. For example, all three of the following procedures will use a codebook called OPINION.COD and a data file called MARKET.DAT.

```
STUDY OPINION
DATA MARKET
CROSSTABS INCOME WITH PREFERENCE
..
BANNERS AGE RACE INCOME WITH PREFERENCE
..
DESCRIPTIVE AGE
..
```

When you use the STUDY command to specify a new codebook, the data file will be changed to the new codebook name automatically. In other words, using the STUDY command overrides all previous STUDY and DATA command specifications. In the following example, the second procedure will use a data file called STUDY-2.DAT, even though the DATA command was used in the first procedure to specify a data file called SENSORY.DAT.

```
STUDY STUDY-1
DATA SENSORY
TTEST PRETEST WITH POSTTEST
..
STUDY STUDY-2
LIST REMARKS
..
```



---

## Continuation Lines

If a command is too long to fit on one line (e.g., a long variable list), StatPac will automatically indent subsequent lines. You can simply continue typing and the word-wrap feature will take care the indentation for you. You can also use an explicit (hard) return to begin the next line. A break between lines should always occur between words or between sets of parentheses. A continuation line is denoted by indenting at least one character (typing at least one space at the start of the line). The following procedure will perform a frequency analysis on eight variables. Since the second line is indented, StatPac will interpret it as a continuation of the previous line.

```
STUDY GOVT
FREQUENCIES AGE RACE SEX INCOME STATUS
CATEGORY HOUSING TRANSPORTATION
..
```

Sometimes, continuation lines can be used to make a procedure easier to read. The following procedure will perform descriptive statistics on three variables. Because the variable names are indented, they will be interpreted as a continuation of the DESCRIPTIVE line.

```
STUDY BUDGET
DESCRIPTIVE INCOME
EXPENSE
PROFIT
..
```

---

## Comment Lines

Comment lines may be included in a procedure file. Their purpose is to allow you to imbed notes within a procedure file. They are especially helpful when reviewing a procedure file that you have not used for a long time. Comment lines will be ignored when performing an analysis. A comment line begins with an apostrophe, or the word REM. There are no restrictions on the text that may be included in a comment line. Comment lines may also use continuation lines. For example, the following procedure contains two comment lines. The second comment also has a continuation line:

```
REM This procedure has two comment lines
STUDY SURVEY12
' This procedure will only use the first 50 records
  for the analysis because the SELECT command is used
SELECT 1-50
FREQUENCIES ATTITUDE
..
```

Comment lines can be useful when debugging a procedure that contains an unknown error. By selectively making each line a comment (adding an apostrophe to the beginning of the line), you can essentially eliminate that line as a possible cause of the error.

You can also turn entire blocks of text in a procedure file into comment lines. To comment a block of text, highlight it, and select Edit / Comment Selected Text. An apostrophe will be placed at the beginning of each highlighted line except .. lines. That way all the lines in one or more procedures can be commented without changing the procedure number. To remove the apostrophes, highlight the text and select Edit / UnComment Selected Text.

---

## V Numbers

Most of the examples in this manual use variable names. However, it is important to note that either variable names or V numbers may be used interchangeably. For example, if AGE is variable twelve in a study, the following two commands would produce identical results:

```
DESCRIPTIVES AGE
DESCRIPTIVES V12
```

---

## Keywords

Designing a procedure with StatPac consists of typing a series of commands. With the exception of comment lines and continuation lines, each line in the procedure will begin with a keyword or analysis command.

Keywords are used to modify an analysis. They may be used in a procedure to change labeling and perform transformations. In fact, they are used for everything except the actual selection of an analysis type. The STUDY and DATA commands are keywords. A listing of the keywords can be displayed by selecting View, Syntax Help. The keyword menu will appear in a window.

**A single procedure can contain many keyword commands, but only one analysis command.** In the following example, the analysis output will have a page heading and title because of the inclusion of two keywords in the procedure.

```
STUDY SYSTEM
HEADING Acme Research, Inc. - System Analysis Study
TITLE Crosstabs between Shift and Efficiency Rating
CROSSTABS SHIFT BY RATING
..
```

---

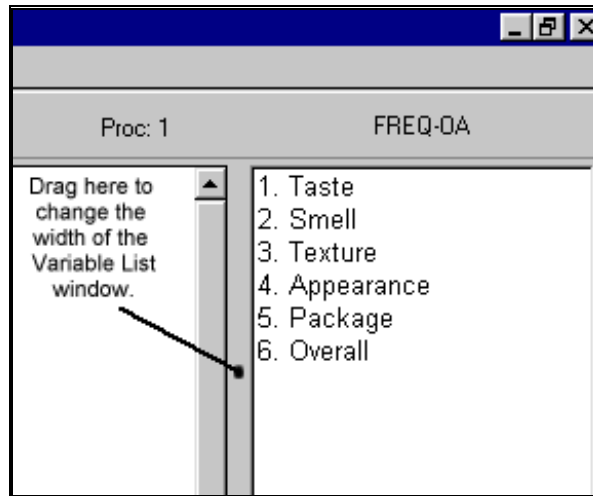
## Analyses

While many keywords can be used in a procedure, only one analysis command can be specified. A listing of the analysis commands can be displayed by selecting View, Syntax Help. The help window will appear.

---

## Variable List

The Variable List window enables you to view and select variables for an analysis. It can be displayed by selecting View, Variable List. The width of the Variable List window can be adjusted by dragging the bar that separates the procedure file text from the Variable List window.



One convenient feature of the Variable List window is the ability to transfer variable names to the procedure text. To select a variable, highlight it in the Variable List window. To select multiple variables, hold down the shift or control key while clicking on the desired variables in the Variable List window.

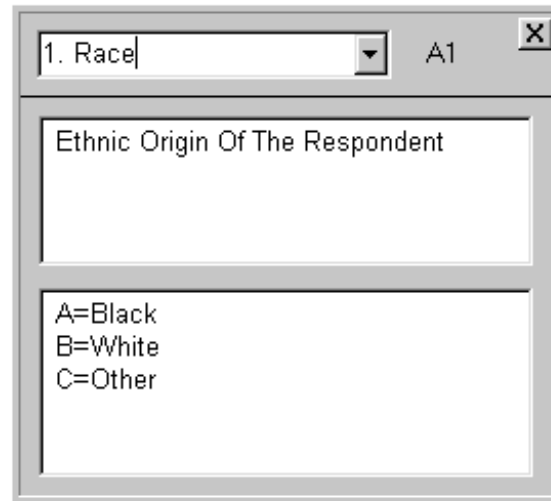
To transfer selected variables from the Variable List window to the text of the procedure file, first select the desired variables in the Variable List window. Then double click in the procedure file where you want the variable names to appear. The highlighted variables in the Variable List window will be copied to the procedure file text, and the variable(s) will be deselected in the Variable List window.

---

## Variable Detail

The Variable Detail window lets you see detailed information about any variable, or change the information for a variable. To display the Variable Detail window, select View, Variable Detail. You can also double-click on a variable in the Variable List window to evoke the Variable Detail window.

All information for a variable can be changed except its format. Changes made in the Variable Detail window (e.g., revised labeling) will be saved in the codebook, and therefore will appear in the analyses.



The current variable displayed in the Variable Detail window can be changed by using the drop-down variable selection or by clicking on the desired variable in the Variable List window.

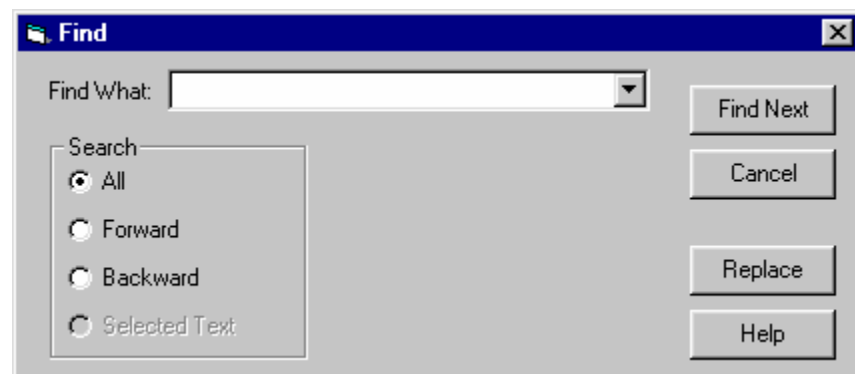
The Variable Detail window can be dragged to any location on the screen. Press and hold the left mouse button anywhere on the gray borders of the window. Drag the Variable Detail window to the desired location and release the mouse button to drop the window at that location.

You can hide the Variable Detail window by selecting View, Variable Detail. Alternatively, click on the X in the top right corner of the Variable Detail window.

---

## Find Text or Variable

Use the Find Dialog window to search for specific text in the procedure file or the results. Select Edit, Find (or use the Ctrl F shortcut) to display the Find Dialog window.

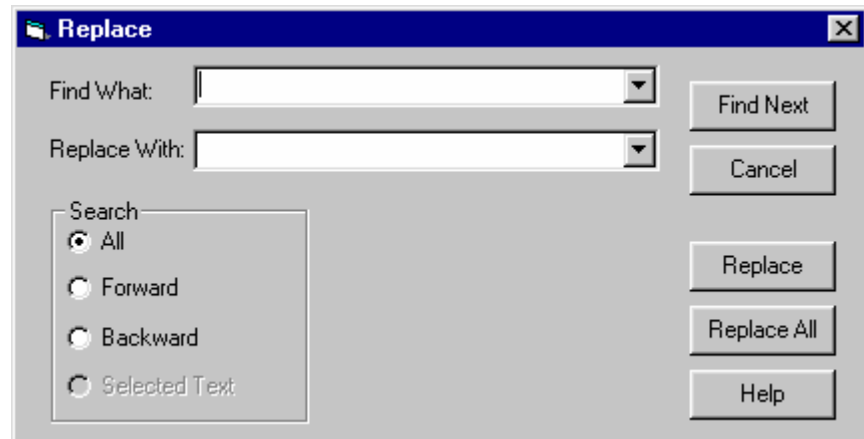


To begin a search, type the search text and click on the Find Next Button. After a search has been started and a match has been found, you can continue the search by clicking on the Find Next Button (or by pressing the [F3] shortcut). Upper and lower case differences will be ignored in the search.

---

## Replace Text

Use the Replace Dialog window to replace specified text in the procedure file or results. Select Edit, Replace (or use the Ctrl H shortcut) to display the Replace Dialog window. Alternatively, you can click the Replace Button from the Find Dialog window.



Upper and lower case differences will be ignored when finding text. However, replaced text will use the exact text typed into the Replace With window.

---

## Options

Options are used to control the analysis. Options allow you to modify the defaults for an analysis; that is, they allow you to customize the analysis parameters themselves. Analysis options may be changed temporarily or permanently. When changed permanently, the current procedure and all future procedures will use the new defaults. When changed temporarily, only the current procedure will use the new options.

Some options are global and apply to all analyses. Other options are specific to the type of analyses being performed. If you select Options when there is no procedure file or when the cursor is in a procedure that does not specify an analysis, only the global options will be displayed. They allow you to set the pitch (point size) for the report, the page margins and paper orientation, the next page number to be printed, zoom factor, and weighting.

The margins are expressed in inches. The paper orientation may be set to OR=P (portrait) or OR=L (landscape). The zoom factor is any easy way to reduce the size of a table so it will fit on one page. Normally, ZF=100 and the printouts will be displayed at 100% their normal size. Setting ZF=80 would display the tables at 80% of their normal size, so more columns would be able to fit on a page. The ecology option may be used to save paper. When EC=Y and you are saving the output to a batch file, all page breaks will be excluded. At the conclusion of the batch run, select System, Current Batch File, to print the file. When running interactively or batch to printer, EC=Y will only suppress page breaks within each task or procedure.

The WT (weighting) option lets you weight the data based on the value of another variable, and the FC (fractional counts) option controls whether the reports will show

integer or fractional counts. They currently apply to all analyses in the Basic Statistics Module.

Option	Code	Setting
Automatic Selection	AS	N
Bottom Margin	BM	.0
Ecology	EC	N
Fractional Counts	FC	N
Left Margin	LM	.5
Orientation	OR	P
Page Number	PG	1
Pitch (Font Size)	PI	10
Right Margin	RM	.5
Top Margin	TM	.5
Weighting	WT	
Zoom Factor	ZF	90

**Automatic Selection**

Sets whether only non-missing data will be selected for analysis.

N=No Y=Yes

OK Cancel Help

To view the options for an analysis, move the cursor to the procedure where the analysis command is specified and select Options. If no analysis is specified in the current procedure, only the global options will be shown.

The options for each analysis are different. If the current procedure contains an options line that changes the default values, the modified values will be displayed in yellow. Any errors in the option line will be displayed in red. To change the option temporarily, simply type the new value for the option. To make a permanent option change, type the new value and add an exclamation point as a suffix. For example, typing Y changes an option to yes for the current procedure only. Typing Y! changes the option permanently so that all future analyses will use the default of Y.

### ***Automatic Selection of Non-Missing Data Option***

In all analyses in the Basic Statistics Module an option can be used to select cases if the desired data is not blank. The AS option is global and will apply to all analyses. For example, if you specify AS=Y! in the first procedure, it will apply to all subsequent procedures until the AS=N! option is specified.

The AS option is a simple way to replace IF-THEN-SELECT command when you want to include only records where the variable being analyzed is not missing.

For example, you wanted to do a frequency analysis of variable 9, and only include records where V9 was not blank, you could do it with any of these three procedures. All three would produce identical results.

```

IF V9 <> "" THEN SELECT
FREQUENCIES V9
..
IF V9 = "" THEN REJECT
FREQUENCIES V9
..
FREQUENCIES V9
OPTIONS AS=Y
..

```

The advantage of the AS option becomes clear when your variable list contains multiple variables. If you had a variable list, such as:

```

FREQUENCIES V9 V11 V15

```

You would have to run each one as a separate procedure in order to select the non-missing data for each variable:

```

IF V9 <> "" THEN SELECT
FREQUENCIES V9
..
IF V11 <> "" THEN SELECT
FREQUENCIES V11
..
IF V15 <> "" THEN SELECT
FREQUENCIES V15
..

```

Now you can use the AS option to accomplish the same thing:

```

FREQUENCIES V9 V11 V15
OPTIONS AS=Y
..

```

When doing this for frequencies, the PB option will no longer have an effect because there will be no missing cases.

The AS option is especially useful when using the LIST command to list comments. Typically, you would use an IF-THEN-SELECT command to list only records that made a comment:

```

IF Comment <> "" THEN SELECT
LIST COMMENT
OPTIONS LB=0 BL=Y
..

```

The same thing can be accomplished with the AS option:

```
LIST COMMENT
OPTIONS AS=Y LB=0 BL=Y
```

```
..
```

When creating Banner tables, the AS option applies to the stub variable or stub variable list (the ones down the side of the page). It will select only records where the stub variable is non-missing.

All three of the the following procedures would select the same data:

```
IF StubVar <> "" THEN SELECT
BANNERS StubVar BY Total BannerVars

..
IF StubVar = "" THEN REJECT
BANNERS StubVar BY Total BannerVars

..
BANNERS StubVar BY Total BannerVars
OPTIONS AS=Y

..
```

### ***Weight and Fractional Counts Options***

The WT option lets you apply non-integer (fractional) weighting to procedures. It is used when the sample differs from known population parameters. To apply case weighting, you must first create a variable that contains a weight.

The following example computes weights for each of three groups and saves the weight for subsequent analyses. The CaseWeight variable will become the last variable in the study.

```
STUDY SEGMENT
NEW (N7) "CaseWeight"
IF GROUP = 1 THEN COMPUTE CaseWeight = 0.4172
IF GROUP = 2 THEN COMPUTE CaseWeight = 0.8735
IF GROUP = 3 THEN COMPUTE CaseWeight = 1.0963
SAVE

..
```

Subsequent procedures could then apply weights to the analyses using the WT option. Parentheses are required around the variable name. Since an exclamation point is used as a suffix, weighting will become the default for all subsequent analyses. In this example, both the frequencies and descriptive statistics procedures would weight the data. If the exclamation point had been excluded, weighting would only be used in the frequencies procedure.

```
STUDY SEGMENT
FREQ V1-V10
OPTIONS WT=(CaseWeight)!

..
DESC V11

..
```



Unlike other options, the WT option (with a ! suffix) only applies to the current StatPac session. If you quit StatPac and restart it, the WT option will be set to N (None). This is done to prevent a potentially serious mistake. For example, suppose you run a procedure file with weighting and then end StatPac. The next day you run StatPac and begin processing a different procedure file. If the WT option was persistent, weighting might inadvertently be applied to the new procedure file when you didn't intend it to be... and worse, you might not realize it.

You can turn weighting on and off by using WT=(VariableName) and WT=N. In the following example, weighting is applied to the first, second procedures, but not the third and fourth procedures.

```
STUDY MYRESEARCH
TITLE Weighted Frequencies for: (#)
FREQ V1
OPTIONS WT=(WeightVariable)!
..
TITLE Weighted Descriptive Statistics for: (#)
DESC V2
..
TITLE Unweighted Frequencies for: (#)
FREQ V1
OPTIONS WT=N!
..
TITLE Unweighted Descriptive Statistics for: (#)
DESC V2
..
```

The FC (fractional counts) option may be set to Y or N. It sets whether the N's (counts) in the reports will be shown as integers or decimal values. The FC option only applies when weighting is used. In unweighted data, the counts will always be integer values (whole numbers).

Weights are easily calculated as the desired percentage divided by the observed percentage (or the desired count divided by the observed count). For example, suppose you know that the population has 55% males and 45% females. This is called a *known population parameter*. Your survey sample, however, has 40% males and 60% females. If the responses to other variables were different for males and females, your reports might present a distorted estimate of the population. Weighting would be used to eliminate the gender sampling error. The weight for males would be 55/40 and the weight for females would be 45/60. In the following example, the first procedure calculates a GENDERWEIGHT variable and saves it. The second procedure uses the WT option to weight the data based on the GENDERWEIGHT variable.

```
NEW (N5) "GENDERWEIGHT"
IF SEX="M" THEN COMPUTE GENDERWEIGHT = 55/40
IF SEX="F" THEN COMPUTE GENDERWEIGHT = 45/60
SAVE
..
FREQ SOMEVARIABLE
OPTIONS WT=(GENDERWEIGHT)
```

### Important User Tip

The first few times you run StatPac for Windows, experiment with the options to find the values that produce the report formatting you want. Rather than setting these options in each procedure, use the exclamation point suffix to make them permanent. After running a few procedures, you'll have configured the default formats for StatPac to produce the reports you most often use.

---

## Load, Save, and Merge Procedure Files

The File selection of the menu allows you to load, merge, and save procedure files. To open a new procedure file, select File, Open, or click the Open Button. To save the current text in a procedure file, select File, Save, or click the Save Button.

To begin a new procedure file, select File, Open, and change the Files of Type to codebooks. Select the codebook and click OK. The STUDY command will be inserted as the first line of the procedure.

Procedure files are always saved with a .pro extension. While we recommend using StatPac for creating and editing procedure files, they are plain ASCII text, and may be edited with any text editor (such as notepad).

If loading a new file, and the current procedure file text has changed, StatPac will check to see if you want to save the current text before abandoning it and loading the new file. Note that **anytime you run a procedure, the entire procedure file will be saved before the procedure is run**. Thus, if you load a new file immediately after running a procedure, it is not necessary to save the current procedure file before loading the new procedure file because it will already have been saved.

To merge the text from a procedure file previously stored on disk into the current text window, position the cursor where you want the text to be loaded and then select File, Merge. The text will be inserted ahead of the cursor.

---

## Print a Procedure File

The current procedure file can be printed by selecting File, Print. Select the procedures you want to print and click OK.

If you choose to specify procedures, you must type the procedure numbers that you want to be printed. Procedure numbers can be separated from each other by commas or spaces. A dash can be used to indicate a range of procedures. For example, the following would print procedures 1, 2, 8, 9, 10 and 15

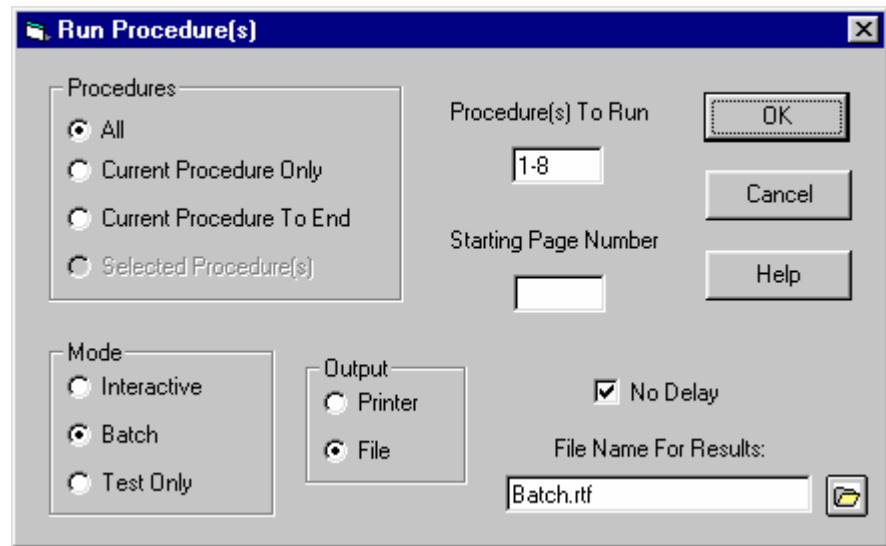
1, 2, 8-10, 15

---

## Run a Procedure File

Click on the Run button to execute the commands in the text window (i.e., to run the analysis). StatPac will give you the option to specify which procedures should be

run, the operating mode, disposition of the output, name of a file to store the output, and the starting page number for the output.



After setting these parameters, click OK to run the analyses.

### ***Procedure(s) To Run***

The "Procedure(s) To Run may be an individual procedure or range of procedures. The default will be the procedure where the cursor was located when the Run button was clicked. If you highlight text before clicking the Run button, the default procedure(s) will be all the procedures that contained highlighted text. A range of procedures may be specified with a dash. To run procedures one through ten, you would type 1-10 in the Procedure(s) to Run field. To run from procedure 5 to the last procedure, you would type 5- in the Procedure(s) to run field. To run a single procedure, simply type the procedure number.

### ***Mode***

The Mode selection allows you to set the analysis to operate interactively, batch, or in the test mode. When interactive is selected, all output will first be displayed on the screen before being sent to its final disposition (printer or file). You will be able to view, edit, print, and save the output; and you must manually tell StatPac when to go on to the next task or procedure.

The Batch mode is similar to the Interactive mode. The difference is that the program will automatically go on to the next task or procedure after showing the results for 3 seconds. During the 3-second display time, you can freeze the screen and view the output of the current task in more detail. You can then continue or cancel the batch run.

The Test mode will simply check the syntax of the selected procedures without actually running them.

When you begin to run an analysis, StatPac will first check the syntax of your procedure(s). The syntax checker will catch all major errors. It is, however, only a syntax checker. It can tell if the syntax is correct, not if the commands will do what you want. It also cannot check for data dependent errors, since these can only be discovered through actual data processing. If a syntax error is discovered, correct the error and re-run the procedure.

## ***Output***

The Output selection will be displayed when processing in the Batch mode. It refers to the final disposition of the output (i.e., where you want the results of the analysis to be sent). You may send the output to the printer or a file. When you choose to send the output to a file, you will also be able to enter a file name for the output. The output will be saved in Rich Text Format.

In the Batch mode, the results of all analyses will first be displayed on the screen for 3 seconds. If you do nothing, the results will then be sent to the Output (printer or file). However, if you temporarily freeze the output by pressing the Pause button, you will have the choice whether to save or print the results.

## ***Starting Page Number***

The Starting Page Number is especially useful when processing in the Batch mode. StatPac will automatically increment the page numbers with each task. If you stop a batch run, or decide to rerun a particular task, you will need to manually set the starting page for the next run.

If the Starting Page Number is left blank, no page numbers will be printed on the output.

The page number will be placed on the page in the location specified in the header or footer template.

## ***No Delay***

When running in the batch mode, the results will normally be shown on the screen for a few seconds before being printed or written to file. This gives you time to review the results before printing or saving them. When the No Delay box is checked, the results will be immediately sent to the output device.

---

# **Results Editor**

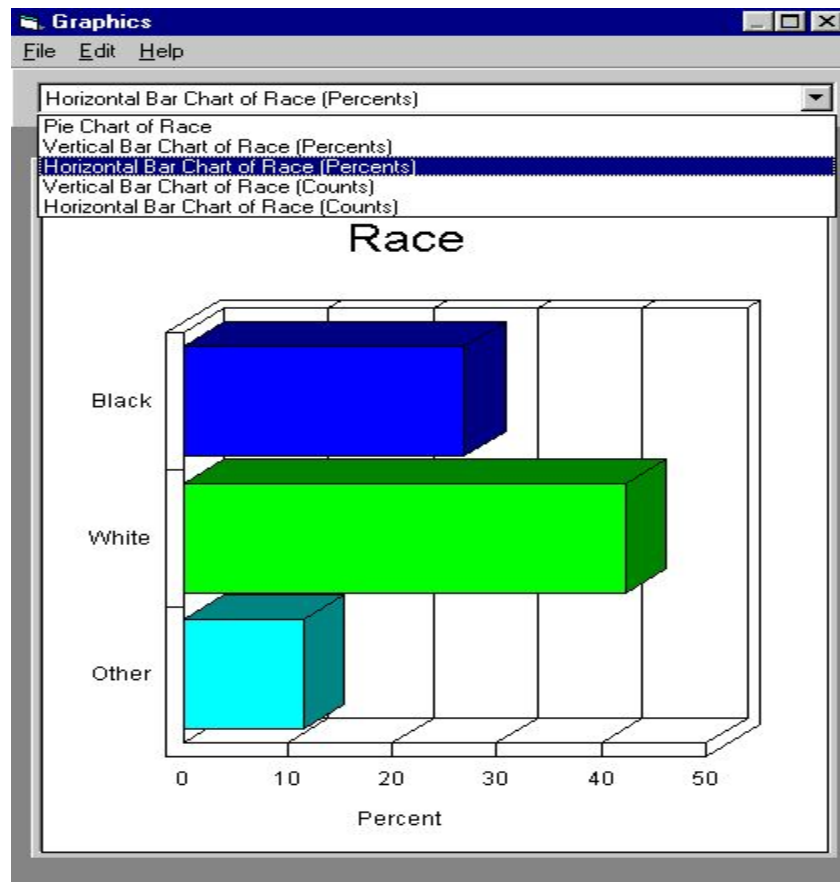
After StatPac has finished processing an analysis, the results will be displayed. (In a batch run, the results will only be displayed for 3 seconds unless you freeze the program with the Pause button in the results editor). The results editor will allow you to examine and edit the results before printing or saving them.

All files saved or loaded with the Results Editor will be in Rich Text Format with a .rtf extension. These files can also be loaded with your word processor.

---

# **Graphics**

Graphics are available when performing frequencies, crosstabs, descriptive statistics, breakdowns, and correlation analyses. A colorful graphics button will be shown on the Results Editor tool bar for these analyses. Clicking the button will let you select and edit the graphs.



To modify the appearance of a graph, select Edit, Legend & Labels. This will give you the opportunity to change any of the text on the graph, including legend information (if there is one). The legend information can be saved by selecting the Save As Default check box. Future graphs with legends will then use the new settings.

**Graphics Labels & Legend** [X]

Use the | character to force a line feed.

Graph Title:  [Font]

Left:

Bottom:  [Font]

Right:

Legend Text:  [Font]

Legend Position: ☐ ☐ ☐ ☐ ☒ ☐ ☐

Legend Space:  %

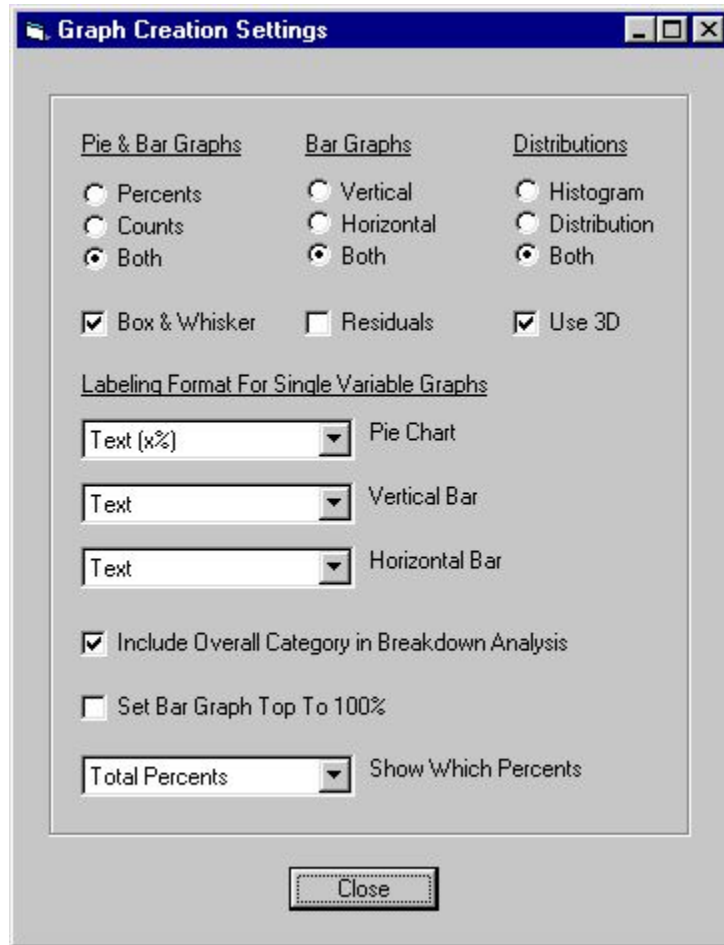
Legend Background Style:  [v]

Tic Labels:  [Font]

Save As Default: ☐

[Close]

The actual creation of the graphs happens while the analysis is being performed. You can control the kinds of graphs that will be created and the labeling methods by selecting Edit, Creation Settings. This option is also available in the Analysis Editor by selecting System, Graph Creation Settings.



After you are satisfied with the appearance of the graph, you can do several things with it:

1. Print the graph immediately by selecting File, Print.
2. Add the graph to the next page of the results by selecting Edit, Copy Graph To Report. When you exit the Graphics Editor and return to the Results Editor, the graph will have been added to the end of the current results.
3. Save the graph as a file (.jpg, .bmp, or .wmf) by selecting File, Save Graph Image.
4. Save the graph in the clipboard by selecting Edit, Copy Graph To Clipboard.
5. Create a tab delimited file of the labels and data used to create the graph (not the graph itself) by selecting File, Save Delimited File.

You can also right click on the graph itself to make changes to the graph.

---

## Table of Contents

When processing in the batch mode a table of contents will be created if page numbering is used. After the batch processing is finished, you can display and print the table of contents by selecting System, Current Table of Contents.

The Title keyword in each procedure of the batch run will be used to create the entry in the table of contents. If a procedure had no title, the table of contents will contain the analysis command line in the procedure.

The current table of contents will be erased and a new table of contents will be started when the starting page is set to 1 in the Run dialog. Setting the starting page number to a value greater than 1 will add to the existing table of contents.

After completing a batch run, select System, Current Table of Contents. Type <Ctrl A> to select the text and <Ctrl C> to copy it to the clipboard. Then open the batch file in MSWord and paste the contents of the clipboard to the new page using <Ctrl V>. Your document will then begin with a table of contents.

You can also create a table of contents that will have hyperlinks into the batch file document, so when you hold down the control key and click on an entry in the table of contents, it will display the referenced page. First, in the System Defaults change HyperlinkTOC = 0 to HyperlinkTOC = 1 and save. Then, when you run a batch file with page numbers, the table of contents will have hyperlinks.

Run your procedures Batch to File. Be sure to specify a starting page number. When completed, select System / Current Table of Contents (opens the Table of Content in MSWord). In MSWord, type Ctrl A and then Ctrl C to select all and copy to the clipboard. Close MSWord. In StatPac, select System / Current Batch File (opens the batch file in MSWord), and in MSWord, select Edit / Paste / Special / RTF. The hyperlinked table of contents will be inserted at the beginning of the document.

---

## Automatically Generated Topline Procedures

A *Topline* contains basic analyses for all the variables in a study. It consists of frequencies, descriptive statistics, and listings of open-ended comments.

When you click on the Analysis button, select System, Automatic Topline.

The report generated by an automatic *Topline* will provide a good summary of the data. If you're just after "answers" and not particularly concerned about labeling, the automatic *Topline* procedures can be run "as is". If you want a "camera-ready" report, you'll want to edit the procedures (especially the Tile and Options commands). Most users will view the automatically generated Topline a solid foundation rather than an end-product.



# Keywords

---

## Keyword Index

These keywords may be used in a procedure to control labeling and perform transformations.

Average	Compute	Count	Data
Difference	Dummy	Footnote	Heading
If...Then...Else	Labels	Lag	Let
Merge	New	Normalize	Options
Recode	Rem	Run	Save
Select/Reject	Sort	Stack	Study
Sum	Title	Weight	Write

---

## Keywords Overview

Keywords are used in a procedure for everything except selecting the analysis type. These words (commands) are recognized by StatPac when used at the beginning of a line. They are used for study and data specifications, labeling output, and data file transformations.

Some keywords will be used often (e.g., STUDY, HEADING, TITLE and OPTIONS). Other keywords will be used only rarely (e.g., LAG, DIFFERENCE, SUM). Most analyses can be designed using only a few keywords.

All procedure files must use the STUDY keyword once in the first procedure. The use of all other keywords is optional and depends upon the situation.

The following is the list of keywords supported by StatPac

STUDY, DATA, SAVE, WRITE, MERGE, HEADING, TITLE, FOOTNOTE, LABELS, OPTIONS, SELECT, REJECT, NEW, LET, STACK, RECODE, COMPUTE, COUNT, SUM, AVERAGE, IF-THEN-ELSE, SORT, WEIGHT, NORMALIZE, LAG, DIFFERENCE, DUMMY, and RUN.

---

# Categories of Keywords

Keywords can be logically divided into four categories. The categories are:

1. Commands for selecting, loading and saving files.

STUDY  
DATA  
SAVE  
WRITE  
MERGE

2. Commands for labeling.

HEADING  
TITLE  
FOOTNOTE  
LABELS

3. Commands for setting analysis options.

OPTIONS

4. Commands for creating new variables and transforming data.

SELECT  
REJECT  
NEW  
LET  
STACK  
RECODE  
COMPUTE  
COUNT  
SUM  
AVERAGE  
IF..THEN..ELSE  
SORT  
WEIGHT  
NORMALIZE  
LAG  
DIFFERENCE  
DUMMY  
RUN

---

## Keyword Help

Most keywords require one or more parameters. These parameters are governed by the syntax requirements of the keyword. Each keyword has its own syntax. To quickly display help for a keyword, select Help, Keywords, and click on the desired keyword to display the help information.

While a procedure can contain only one analysis specification command, it can have many keyword commands. Keywords are often used in combination with each other to form more complex procedures. In the following procedure, the first four lines are keyword commands and the fifth line is an analysis specification command.

```
STUDY ATTITUDES
HEADING StatPac Analysis of the Attitude Survey
TITLE Overall Attitude Broken Down By Sex
OPTIONS PF=NRCT GR=Y
CROSSTABS SUMMARY BY SEX
..
```

---

## Ordering Keywords

Keywords can be used in combination with each other to perform virtually any selection or transformation. The order of the keywords in a procedure may or may not be important depending on the individual procedure.

Generally, it is important to consider the order of keywords whenever a procedure involves more than one transformation. If the results of one transformation are dependent upon another transformation, then proper order is imperative.

For example, in the following procedure three COMPUTE keywords are used to calculate values for three different variables. Since all of the computations are independent from each other (one doesn't depend on the results of another one), the order of the keyword commands doesn't make any difference. The following COMPUTE commands could be specified in any order.

```
STUDY SCIENCE
COMPUTE VAR_1 = 0
COMPUTE VAR_2 = 1
COMPUTE VAR_3 = 2
SAVE
..
```

When one keyword command depends upon the result of a different keyword command, the order of the commands is important. In the following example, each COMPUTE command uses the result of a previous command to perform its computation. Therefore, the order of the commands must be correctly specified.

```
STUDY SCIENCE
COMPUTE VAR_1 = 0
COMPUTE VAR_2 = VAR_1 + 1
COMPUTE VAR_3 = VAR_2 + 1
SAVE
..
```

---

## Global and Temporary Keywords

Only three keywords are global (STUDY, DATA and HEADING). Once used, subsequent procedures will use the same study, data file and/or page heading. These parameters can be changed in any procedure and subsequent procedures will use the new parameters.

All other keywords are temporary and apply only to the procedure in which they appear. In the following example, the first procedure will use only the first 1000 records for the descriptive statistics analysis. The second procedure, however, will use all the records because the SELECT keyword only applies to the procedure in which it appears. The STUDY and HEADING (global keywords) will be the same for both procedures.

```
STUDY CROPS
HEADING Summary Statistics
SELECT 1-1000
DESCRIPTIVE YIELD
..
DESCRIPTIVE COST
..
```

---

## Permanently Change a Codebook and Data File

Two keywords (SAVE and WRITE) allow you to make permanent changes to studies and data files. Any transformations in a procedure will become permanent if the SAVE command is included anywhere in that procedure. Permanent means that the study information and data file are **irretrievable** changed to reflect the transformations requested by the keywords. If you had a study of 10,000 records, the following procedure file might result in a **huge loss of data** and be devastating!

```
STUDY MARKET
SELECT 1001-2000
SAVE
..
```

Note that the inclusion of the SAVE keyword makes the selection permanent. That is, after executing this procedure, the data file will contain only 1000 records. All other records will have been eliminated by the SELECT command. The implications of this are obvious. **Don't use the SAVE command unless you have a backup of the study information and data file.**

We recommend that all procedure files begin by saving duplicates of the codebook and data, and subsequent procedures use the duplicates (leaving the original codebook and data intact). If the previous example were changed to the following, the MARKET codebook and data file would not be affected by the use of the SAVE command in the second procedure.

```
STUDY MARKET
WRITE MARKET1
..
```

```
STUDY MARKET1
SELECT 1001-2000
SAVE
..
```

---

## Backup a Study

We **strongly** recommend making a backup of all study information and data files before beginning any analyses. If you have a backup, you will always be OK and you won't have to worry about saving erroneous transformations. If you don't have a backup, and you accidentally save an unwanted transformation, nothing can be done to recover the data.

Making a backup of a study is easy... you just have to remember to do it. Please, if you plan to use the SAVE command, make a backup first.

There are three basic ways to backup StatPac information:

1. Use Windows Explorer to copy the desired files to another folder or drive.
2. Select Data, Backup to create a backup folder and copy all files from the current data folder to the new folder
3. Run a procedure using the WRITE command to save a copy of the codebook and data file.

To make a backup of a codebook and data file, you need to run a two-line procedure. The first line specifies the codebook name (and its associated data file) and the second line uses the WRITE command to specify the study name for the backup files. For example, if we have a study called SURVEY, the following procedure will create a backup called B-SURVEY. The backup will contain both the study information and the data file.

```
STUDY SURVEY
WRITE B-SURVEY
..
```

The WRITE command is used to specify the name of the backup files. The name may include a drive or path. For example, if you wanted to store a backup copy on a diskette in the A drive, you would run the following procedure.

```
STUDY SURVEY
WRITE A:\B-SURVEY
..
```

The previous examples would not create a backup of the data entry form. The form itself is only used for data entry and editing, which is normally finished by the time you begin running analyses. Therefore, the form is not altered by any transformation or analysis. You can, however, manually make a backup of a form.

We suggest that you start all procedure files with a two-line procedure that creates a duplicate codebook and data file, and in the second procedure, begin using the new files, thus leaving your original file unchanged. In this example, the first procedure reads the SURVEY codebook and data file, and creates a new codebook and data file called SURVEY2. The second procedure says to use the SURVEY2 codebook and data for the second and all subsequent procedures. Thus, if any serious mistakes are

made in transforming data, you could easily revert back to the original codebook and data by rerunning the first procedure.

```
STUDY SURVEY
WRITE SURVEY2
..
STUDY SURVEY2
(begin your procedures here)
..
```

As you become more experienced with StatPac, you may include other lines in the first procedure. You might want to create a few new variables that you intend to use later on in the procedure file. For example, you might want to always begin your procedure files by creating an absolute record number variable (useful for identifying bad data) and a net variable that can be used in banner tables. The following might be the beginning of the procedure file:

```
STUDY MYSTUDY
COMPUTE (N5) REC = RECORD
NEW (N1) "NET" Totals
LABELS NET (=)
WRITE TEMP-MYSTUDY
..
STUDY TEMP-MYSTUDY
(begin your procedures here)
```

---

## STUDY Command

The STUDY command is used to specify the name of the codebook and data file being analyzed. It must be specified in the first procedure. Subsequent procedures will use the same study name unless another STUDY command is used to change it. The STUDY command may only be used once in any given procedure.

The syntax of the command is as follows:

```
STUDY <File name>
```

The file name (or study name) may include a path specification. If no path is specified, StatPac will assume that the study resides in default data subdirectory. **No extension should be used when specifying a study name.**

All the following are examples of the proper use of the STUDY command:

```
STUDY MARKET
STUDY C:\DATA\BOOK
STUDY A:SURVEY
```

---

## DATA Command

The DATA command may be used to specify the name of the data file to be analyzed. It is only used when the data file name is not the same as the study name. The DATA command may be used only once in a given procedure.

The syntax of the command is as follows:

**DATA <File name>**

The file name may include a path specification. If no path is specified, StatPac will assume that the data file resides in the default data subdirectory. It is not necessary to use the extension .DAT when specifying the data file name; StatPac will automatically use this extension for all data files.

Using the DATA command changes the data file for the current and all subsequent procedures. The data file will continue to be analyzed until changed by another DATA or STUDY command.

All the following are examples of the proper use of the DATA command:

**DATA RAWDATA**

**DATA A:RESULTS**

**DATA C:\STATPAC\HEALTH\SURVEY**

---

## SAVE Command

The SAVE command is used to make all transformations in a given procedure permanent.

There are two forms of syntax for the SAVE command. In the first form, the SAVE command is specified on a line by itself. When used this way, all transformations will be saved to the current codebook and data file. As a safety precaution, do not use the SAVE command in this way unless you have a back up of the study information and data file.

**SAVE**

The other form of the SAVE command lets you save the codebook and data to a different file. This form of the command is functionally identical to the WRITE command except that a variable list may not be specified. If a path is not specified as part of the output file name, the default data subdirectory will be assumed. If the file name contains spaces, you must enclose the file name in quotes or parenthesis.

**SAVE <File name>**

For example, the following procedure creates a new five-character variable called AVG (the average of eight test scores), and stores this as a permanent part of the codebook and data file.

```
STUDY SCORES
AVERAGE (N5.1) AVG = SCORE1 - SCORE8
SAVE
..
```

After running this procedure, the codebook SCORES.COD and data file SCORES.DAT will have one more variable than they had before the procedure was run. The SAVE command makes the new variable permanent.

As another example, consider the following two procedures. The first procedure uses the SORT command to sort the data file in ascending order by last name. Because the SAVE command is used, the data file will be saved in sorted order. Also notice that the save command specifies a file name, so the sorted data will be saved to the new file. The second procedure begins with the STUDY command to tell StatPac to use the new file and then prints a listing of three variables. The listing will be in sorted order because this is now the way the new data file is stored.

```
STUDY NAMES
SORT (A) LAST_NAME
SAVE "My New File Name"
..
STUDY My New File Name
LIST FIRST_NAME LAST_NAME PHONE
..
```

If you specify a file name with the SAVE command, your original codebook and data remains untouched, and you never have to worry about making a mistake. Even when you do not specify a file name, a mistake doesn't necessarily mean you should panic. The only transformations that are irreversible are those that change the original variables. As long as your raw data is intact, you should be able to recover from any mistake.

---

## WRITE Command

The WRITE command is used to save transformed data in a file or write only selected variables to a file. Unlike the SAVE command (that saves the transformations in the original file), the WRITE command is usually used to save the transformed data in a different file.

The syntax for the WRITE command is:

```
WRITE <File name> <Variable list>
```

The WRITE command actually creates a new codebook and data file called a *subfile*. The subfile is just like any other study. The difference between the original study and the subfile is that the subfile reflects all transformations performed in the procedure.

While the SAVE command saves all transformations in the original file (unless a filename is specified), the WRITE command always creates a new codebook and data file and saves all transformations in the new files. The <Variable list> parameter is used to control what variables should be written to the subfile. When the <Variable list> is not specified, all variables will be included in the subfile.



The following procedure would use a codebook called NAMES (and an implied data file called NAMES). The procedure will sort the file by last name and save the sorted study as a subfile using a new study name SORTNAME. All variables from the original file will be included in the subfile.

```
STUDY NAMES
SORT (A) LAST_NAME
WRITE SORTNAME
..
```

In a similar example, this procedure file selects every tenth name from the original file (NAMES) and creates a new study (subfile) called MINIFILE. The subfile will contain the same variables as the original study, however, there will be only one-tenth as many records. The second procedure uses the STUDY command to access the subfile and list three selected variables.

```
STUDY NAMES
COMPUTE (N7.1) REC = RECORD
COMPUTE (N7.1) IREC = REC/10
COMPUTE IREC = INT(IREC)*10
IF REC = IREC THEN SELECT
WRITE MINIFILE
..
STUDY MINIFILE
LIST FIRST_NAME LAST_NAME PHONE
..
```

The original data file (NAMES.DAT) is called the input file. It is the input data for the transformation procedure. The transformed data file (MINIFILE.DAT) is called the output file. It contains the output from the transformation procedure. The second procedure uses the STUDY command to access the output file from the first procedure. MINIFILE is now, in effect, its own study. The procedure created a MINIFILE.COD codebook and MINIFILE.DAT data file.

When the output file name is different from the input file name, the input file will remain intact, and the new output file will contain the transformed data. When the input file name and the output file name are the same, the WRITE command functions identically to the SAVE command. In the following example, the output file will write over the input file and the original (pre-transformed) data will be lost. Unless you have an up-to-date backup, we recommend specifying a unique file name when using the WRITE command to eliminate the possibility of losing data.

```
STUDY FOOD
COMPUTE RISK = LOG( RISK )
WRITE FOOD (causes a loss of the original RISK data)
..
```

The output file name parameter must be specified whenever the WRITE command is used. If a path is not specified as part of the output file name, the default data subdirectory will be assumed. If the output file name contains spaces, then you must enclose the name in quotes or parenthesis. For example:

WRITE ANOTHER FILE	Wrong
WRITE "ANOTHER FILE"	Correct
WRITE (ANOTHER FILE)	Correct

When a procedure does not create any new variables or change any labels, you can use the WRITE command to save a new data subfile without saving the codebook. It is not necessary to resave the information if none of the study information has been changed by transformations. To save a data file without saving the study information, add a .DAT extension to the file name parameter. Subsequent procedures could use the same (original) study information, but would require a DATA command to access the new data subfile. In the following example, the first procedure selects males from the file and writes a data subfile consisting of just males. The second and third procedures access this subfile with the DATA command and perform analyses on it. The analyses will only include those records selected in the first procedure. Note that the DATA command is required only in the second procedure because it remains in effect until changed by a STUDY command or another DATA command.

```

STUDY STATUS
IF SEX = "M" THEN SELECT
WRITE MALES.DAT
..
TITLE Income Statistics for Male Respondents
DATA MALES
DESCRIPTIVE INCOME
..
TITLE Housing Information for Male Respondents
FREQUENCIES HOUSING
..

```

The WRITE command may be used to create a subfile of selected variables simply by specifying which variables are to be contained in the subfile. The <Variable list> parameter allows you to control which variables (and in which order) will be written to the subfile. All the previous examples did not specify a variable list so StatPac would write all the existing and newly created variables to the subfile. In the following example, two variables (OVERALL and LASTYEAR) will be written to the new subfile SUMMARY.

```

STUDY ATTITUDES
WRITE SUMMARY OVERALL LASTYEAR
..

```

The variable list may be specified as individual variables, a range of variables, or a combination of the two. It may include both variable names and V numbers. Either commas and/or spaces may be used to separate variables. Continuation lines may be used for long variable lists. All of the following WRITE commands contain valid variable lists:

```

WRITE SUMMARY V1-V3 V10 V11 V15
WRITE SUMMARY V1, V2 ,V3, V10, V11, V15

```

**WRITE SUMMARY AGE INCOME V9-V14 V97**  
**WRITE SUMMARY TEST1-TEST9 V4-V7 V1-V3**

The WRITE command may also be used to change the order of the variables. When the new output file is created, variables will be written in the same order as specified in the variable list. Thus, it is easy to restructure the order of variables in a codebook and data file.

For example, suppose you have finished creating a codebook and data entry form. As you begin to enter data, you realize that you forgot to include a variable in the study design. If you have only entered a small amount of data, the easiest way to correct the problem is to:

- 1) delete the data file
- 2) add the new variable to the codebook and form using Study Design
- 3) begin entering data again.

However, if you have already entered a large amount of data, you may not want to delete the data you have already entered. In this case, the WRITE command can be used to correct the problem.

In this example, you might want to add a new open-ended response variable immediately following variable 55 in a study called MYSTUDY. The codebook currently contains 88 variables. The following procedure creates a new variable called OTHER, and then resaves the codebook and data file inserting the new variable into the middle of the study. After running the procedure, the study and data file will contain 89 variables. The new variable (OTHER) will be blank for all existing records in the data file. The data entry form is not automatically updated, so you will also need to add the new variable to the form before using it to edit or enter data.

**STUDY MYSTUDY**  
**NEW (A50) "OTHER" Other Brand Specified**  
**WRITE MYSTUDY V1-V55 OTHER V56-V88**

..

The WRITE command can also be used to increase the length of an open-ended response variable. Suppose that variable 12 in the study is called CITY, and it is 15 characters in length (i.e., the format is A15). After entering a large number of records into the data file, you come across a city name that is 18 characters in length, and you want to increase the length of the CITY field from 15 to 20 characters. First, run the following procedure to create a new 5 character dummy variable and write it as the 13th variable in the study (immediately following the CITY variable). This will insert 5 blank spaces after the current 15 character CITY field. Then, go to the study design, delete the dummy variable and change the format for the CITY field to A20. Again, the data entry form is not automatically modified by this procedure, and therefore, it needs to be changed to reflect the new field length for the CITY variable.

**STUDY MYSTUDY**  
**NEW (A5) "DUMMY"**  
**WRITE MYSTUDY V1-V12 DUMMY V13-V88**

..

---

## MERGE Command

The MERGE command is used to merge a study and data file into the current study and data file. It is usually appears as the only line in a procedure. The syntax of the MERGE command is:

**MERGE <filename>**

The <filename> is the name of the study and data file that you want to merge into the current study. After running the procedure, the variables (and data) from the merge file will be added to the end of your original variables. That is, after running the procedure, the number of variables in your original codebook will increase. The new variables will appear in the Variable List window.

Both the current study and the file to be merged must have the same number of records and they must be in the same order. In other words, record one in the current data file is the same respondent as record one in the data to be merged.

The MERGE command does not automatically delete the files after the merge is completed. The study and data will be merged into your current study, but they will also remain on disk in their original form. If you want to delete the study and data after the merge is completed, add a /K to the end of the <filename>. For example, the following MERGE command would add the variables and data from a file called DEMOGRAPHICS to the current study (RESEARCH), and then delete the DEMOGRAPHICS codebook and data after the merge has been successfully completed.

**STUDY RESEARCH**  
**MERGE DEMOGRAPHICS /K**  
**..**

---

## HEADING Command

The HEADING command is the method used to place a page heading on the printouts. It will appear in the top left corner of all printouts. The syntax for the HEADING command is:

**HEADING <Page heading>**

The use of the HEADING command is optional. If it is not included in a procedure file, the default page heading will be used.

Like the STUDY command, the HEADING command is usually specified in the first procedure. Subsequent procedures will use the same heading as the previous procedures unless the HEADING command is used to change the page heading.

Examples of the HEADING command might be:

**HEADING Family Planning Attitudes Study - 1999**

HEADING StatPac Inc. - Computer Software Division  
HEADING CLASSIFIED INFORMATION - SECURITY CLEARANCE  
REQUIRED

There are no restrictions on the content of the heading. Both upper and lower case characters may be used. The heading may be printed as a blank line if the keyword HEADING is used by itself with no other characters on the line.

The HEADING command may be used only once in any given procedure. All tasks in that procedure will use the same heading. Subsequent procedures will, by default, use the last heading specified, or, the heading may be changed by using the HEADING command to reassign a new page heading to the output.

---

## TITLE Command

The TITLE command is used to place a procedure or task title on the printouts. It will appear in the top left corner below the page heading.

The syntax for the TITLE command is:

TITLE <Procedure title>

The use of the TITLE command is optional. Unlike the HEADING command, it applies to only the procedure in which it appears. If a title is not specified for a procedure, no title will appear on the output.

There are no restrictions on the content of the title. Both upper and lower case characters may be used. The title may be printed as a blank line if the keyword TITLE is followed by two or more spaces, with no other characters on the line.

Examples of the TITLE command might be:

TITLE First Run of the Data - Descriptive Statistics  
TITLE Frequency Analysis of Product Acceptance Questions  
TITLE Comparisons of Males & Females

There is a special feature that may be used in the TITLE command. The three-characters (#) may be substituted in the title in place of the variable name. The title on the printout will contain the variable label instead of the (#) symbols. For example, the following procedure would produce three different titles. Each title will substitute the correct variable label in place of the (#) symbols.

STUDY SURVEY  
TITLE Frequency Analysis of (#)  
FREQUENCIES AGE RACE SEX  
..

This is especially useful when creating banner tables. When running the same banners on a series of stub variables, it is often desirable to place the variable label for the stub variable on the top of the page. The following commands would run 25 different banner tables (one per page) and the title for each table would be the

variable label from on the stub. The CO option turns off compression so that only one stub variable gets printed per page.

```
STUDY SURVEY
TITLE (#)
BANNERS V1-V25 BY AGE RACE SEX
OPTIONS CO=N
..
```

---

## FOOTNOTE Command

The FOOTNOTE command may be specified in any procedure to place a footnote at the bottom of each page of output. Only one FOOTNOTE command may appear in a procedure. The syntax for the FOOTNOTE command is:

```
FOOTNOTE <Page footnote>
```

The use of the FOOTNOTE command is optional. If it is not included in a procedure file, no footnote will be printed. There are no restrictions on the content of the footnote. Both upper and lower case characters may be used.

Examples of the FOOTNOTE command might be:

```
FOOTNOTE This Analysis Was Produced By StatPac Inc.
FOOTNOTE Note: Includes all data collected through 1991.
```

The functionality of the FOOTNOTE command is like the HEADING command, where once specified, the footnote will apply to all subsequent procedures. To cancel a previously specified footnote, use the FOOTNOTE command without any footnote text (a blank footnote).

---

## LABELS Command

The LABELS command may be used to assign value labels to a newly created variable or to change the value labels for an existing variable. It may also be used to change a variable label. The syntax of the command to change one or more value labels is:

```
LABELS <Variable list> (<Code>=<Label>)(<Code>=<Label>)...
```

The syntax of the command to change one or more variable labels is:

```
LABELS <Variable list> = <New Variable Label>
```

Generally, when the LABELS command is used to change a variable label, only a single variable is specified (although a variable list could be used to change the variable labels for a series of multiple response variables). This example changes the variable label for variable five to "How do you feel about the program?". Note that the variable label text is not enclosed in parentheses or quotes.

```
LABELS V5 = How do you feel about the program?
```

When used to change value labels, the LABELS command is often used in conjunction with the RECODE command. For example, let's say a survey asked the respondent's age (AGE). For our purposes, it is sufficient to report the age as either "under 21" or "21 and over". The RECODE command would be used to recode the data, and the LABELS command would be used to assign value labels to the new categories:

```
STUDY VOTING
RECODE AGE (LO-20=1)(21-HI=2)
LABELS AGE (1=Under 21)(2=21 and Over)
FREQUENCIES AGE
```

```
..
```

Continuation lines are allowed in the LABELS command. For example, if AGE were to be divided into five groups, the LABELS command could be entered as:

```
RECODE AGE (LO-20=1) (21-30=2) (31-40=3) (41-50=4) (51-HI=5)
LABELS AGE (1=Under 21) (2=21-30 Years) (3=31-40 Years) (4=41-50
Years) (5=Over 50 Years)
```

```
..
```

When used to change value labels, two restrictions apply to the use of the LABELS command. The first is that some discretion should be used in the length of the value labels. Excessive value labels will make printouts difficult to read. A good guideline is to limit value labels to about 30 characters. The second restriction is that the code on the left of the equals sign must not contain more characters than the field width.

Value labels being assigned to a new or existing variable will be temporary and apply only to the current procedure. The value labels can be made permanent by using the SAVE or WRITE commands.

The LABELS command may be used more than once in a procedure and it may be used to assign new labels to more than one variable. For example, the following procedure file uses the LABELS command to assign new value labels to ten consecutive items on a questionnaire, and to another variable called OPINION. Because the SAVE command is also specified in the procedure, the new labels will become a permanent part of the study information, replacing any previous labels.

```
STUDY SURVEY
LABELS ITEM1-ITEM10 (A=Very much) (B=Somewhat) (C=Not at all)
LABELS OPINION (1=Positive) (2=Undecided) (3=Negative)
SAVE
```

```
..
```

When using the LABELS command to change a single value label or add new value labels, it is not necessary to retype all the value labels. Adding an exclamation mark to the end of the LABELS command instructs StatPac to update the current labels rather than completely replace them.

For example, if you want to show the no-responses on a printout, even though there is not a value label for the no-responses, just add an exclamation point to the end of the line and StatPac will update the existing value labels. The following command would add a "No response" value label to variable one. The value labels that already exist for variable one would remain intact.

**LABELS V1 ( =No response)!**

In another example, suppose two of the value labels for an income variable are: "1=Less than \$10,000 per year" and "5=More than \$35,000 per year". After reviewing a banners table, you decide that it would look better if the value labels were abbreviated. The following line could be used to change the value labels for codes 1 and 5 without affecting the value labels for codes 2, 3, and 4.

**LABELS V1 (1=<10,000) (5=>35,000)!**

Any value label in the LABELS command may include a vertical bar to force a new line in the printout. (The vertical bar appears as two stacked vertical bars on the keyboard). This is especially useful in banner tables when you want to force the location of a break in the banner heading. For example, the following value label will cause the word "Responsibility" to print on two lines with a hyphen after the letter "n":

**LABELS V1 (1=Respon-|sibility)**

---

## OPTIONS Command

The OPTIONS command is used in conjunction with any of the analysis commands. Its purpose is to specify the computational parameters used to control the analysis, and to select the kinds of printouts desired.

The default parameters for each analysis are stored in the StatPac.Ini file. When you run an analysis without specifying any options, the analysis will be run using the default parameters. The OPTIONS command is simply a way to override or change these settings. If the OPTIONS command is excluded, the values in the table will be used for the analysis.

The analysis editor allows you to display and modify the options for the current procedure by selecting Options. The options will appear, and you will be able to modify them. You can also type options directly on the OPTIONS line in a procedure.

Each analysis has its own options. After selecting Options, the options and their default values will be displayed. You can temporarily change any options by simply entering the desired value. The change will be temporary in that it will only apply to the current procedure. To permanently change an option, add an exclamation point



suffix (!) to the desired value. The default value for the option will be permanently changed so that the new value becomes the default.

The following information is only necessary if you choose to manually enter the options.

All of the analysis options are designated by two-letter codes. Use one or more spaces to separate the options. The format for the OPTIONS command is:

```
OPTIONS <Code>=<Value> <Code>=<Value> <Code>=<Value>...
```

To set an option, type the two-letter code followed by an equals symbol and the value you want to give the option. For example, to set one decimal place on the report (option DP), you could type:

```
OP DP=1 (Note: OPTIONS can be abbreviated as OP)
```

If there are too many option codes to fit on a single line, continue typing and let the automatic word-wrap take care of indenting the continuation line. If you use a hard return at the end of a line, make sure that the break between lines occurs between two options so that no option specification is divided by the break. For example:

```
OPTIONS DS=N RS=Y RC=N MS=Y ST=Y FO=2 PS=Y
      DP=1 AC=Y PR=Y
```

Alternately, multiple options statements may be specified in the same procedure. They will be interpreted as if they were one option line. The previous options line could have been specified as:

```
OPTIONS PI=12 DS=N RS=Y RC=N MS=Y ST=Y FO=2 PS=Y
OPTIONS DP=1 AC=Y PR=Y
```

Generally, options only apply to the procedure in which they appear. If an exclamation point is added as a suffix to the option, it will become the default for all future analyses unless changed by another option. In the following example, the OPTIONS line in the first procedure sets the default for the percentage base to the number of cases. Because, the PB option ends with exclamation point, the second procedure (and all subsequent procedures) will continue to use the same percentage base (i.e., PB=N).

```
STUDY MYSTUDY
FREQ V1-V50
OPTIONS PB=N!
..
FREQ V60-V70
..
```

---

# SELECT and REJECT Commands

The SELECT and REJECT commands are used to create a subset of the data that consists of just some of the records from the original file. The commands are temporary and apply only to the procedure in which they are specified.

You can use the SELECT or REJECT commands to select or reject by record number range or by other criteria. The syntax for the SELECT and REJECT commands are identical. Records will be selected or excluded from the procedure based upon the selection or rejection criteria.

The format for the SELECT and REJECT commands to select by record number range is:

**SELECT <Low record # - High record #>**

**REJECT <Low record # - High record #>**

Type the keyword SELECT followed by the record number range. For example, to select just the first 50 records from your data file, you would type:

**SELECT 1-50**

This would cause the first 50 records to be selected for further processing.

You could exclude a single record from an analysis (say record 25) with the following command:

**REJECT 25**

You may exclude the high record number if you want the entire last part of the file to be used in the procedure. For example, the following SELECT command would skip the first 74 records from the data file. The procedure will use all the records from 75 on.

**SELECT 75-**

Similarly, the following command would select only the first 50 records for an analysis (records 51 to the last record would be rejected):

**REJECT 51-**

When the SELECT or REJECT command is specified, subsequent procedures will use the full data set (not just the selected records). In the following example, the first procedure lists 50 records, while the second procedure lists the entire data file.

```

STUDY QUESTION
SELECT 1-50
LIST ID OPINION GROUP
..
LIST ID OPINION GROUP
..

```

The WRITE command is used to create a subfile of the selected data so that subsequent procedures could access the subfile. When the WRITE command is not specified, the selected records will be used for all the tasks in the procedure, but will not be written to a permanent subfile.

**Never use the SAVE command in the same procedure as the SELECT command unless you have a backup of the study information and data.** Doing so will eliminate records from your data file, and they will not be recoverable without a backup.

An alternate form of the Select command may be used to randomly select a specified number of records from the data file. The syntax is:

```

SELECT <Number of records> /R

```

For example, the following command would randomly select 30 records from the data file and write a new codebook and data file called RND:

```

SELECT 30 /R
WRITE RND
..

```

Again, do not use the SAVE command in the same procedure as the SELECT command. Doing so would erase the original data file and replace it with a data file containing the selected records (resulting in a loss of data). Instead, use the WRITE command to create a new data file that contains just the selected records.

Each time a procedure with the SELECT command and the /R option is run, a new set of random records will be selected. Running the above procedure multiple times would produce a different set of random records with each run. This can be altered by setting an explicit random number seed.

In the System Defaults File (StatPac.ini), the default for the RandomNumberSeed parameter is blank and will be shown as:

```

RandomNumberSeed =

```

A blank random number seed means that a new set of random numbers will be created each time the random number generator is evoked. If you change RandomNumberSeed to any numeric value (File>Open>System Defaults File), StatPac will use the same random number sequence each time it's run. The number itself determines the random sequence (which you cannot control), so it really

doesn't matter what number you enter, although different numbers will produce different random number sequences. A given number will always produce the same sequence.

```
RandomNumberSeed = 25
```

Another form of the SELECT and REJECT commands is used in conjunction with the IF-THEN command. For example, to select just males for an analysis, you would enter the command:

```
IF SEX="M" THEN SELECT
```

A record will be selected if the criteria is met (i.e., if SEX is equal to M). The following would also select just the males by eliminating the females:

```
IF SEX="F" THEN REJECT
```

The general form of this command is:

```
IF <Statement> THEN SELECT
```

The spaces in the syntax are mandatory. There must be at least one space after the IF and at least one space on each side of the THEN. Spacing within the <Statement> portion of the command doesn't matter.

If the <Statement> portion of the command is true for a given record, that record will be selected and written to the output file and/or included in the analysis. If the <Statement> portion is false for a record, it will be skipped (omitted from the procedure).

The quotation marks around the code to be selected are mandatory for alpha-type data only. For numeric-type variables, the quotation marks are unnecessary. The following procedure would first select students that had a grade point average of 3.5 or higher, and then perform two frequency analyses using the selected records.

```
STUDY STUDENTS
IF GPA >= 3.5 THEN SELECT
FREQUENCIES SEX RACE
..
```

Similarly, the following analyses would only be performed on students who had at least a 2.0 grade point average:

```
STUDY STUDENTS
IF GPA < 2.0 THEN REJECT
FREQUENCIES SEX RACE
..
```

The SELECT Command can perform an Nth record selection when used in combination with compute commands. For example, the following procedure would list every tenth record. Note that the integer function is used to check for the record numbers that are evenly divisible by ten. Also note that the REC variable was computed so that the listing would show the record number from the original data file instead of its sequence number in the selected subset.

```
STUDY MARKET
COMPUTE (N5) REC = RECORD
COMPUTE (N5) INTEGER = INT(REC/10)
COMPUTE (N5) REAL = REC/10
IF INTEGER = REAL THEN SELECT
LIST REC NAME PHONE
..
```

The SELECT or REJECT command may be used to exclude missing data from an analysis by selecting only non-blank records. For example, either of these two commands could be used to select non-blank records for an analysis of an attitude question, you would enter the command:

```
IF ATTITUDE <> " " THEN SELECT
IF ATTITUDE = " " THEN REJECT
```

Notice that a space is used to indicate missing data. The SELECT command says to select all records where ATTITUDE is unequal to a blank. The REJECT command says to exclude all records where ATTITUDE is equal to a blank. The result of either command will be the same. Records with non-missing data will be selected. When selecting missing data, quotation marks are required regardless of whether the variable is alpha or numeric. There does not have to actually be a space between the quotation marks (i.e., two quotation marks together would accomplish the same thing).

The SELECT and reject commands are often used in conjunction with the LIST command to list open-ended comments. The purpose is to eliminate records where the respondent made no comment. Both of these procedures would produce identical printouts.

```
IF COMMENT <> " " THEN SELECT
LIST COMMENT
..
IF COMMENT = " " THEN REJECT
LIST COMMENT
..
```

The SELECT and REJECT commands may also be used in combination with AND and OR relational operators to select or reject records based on multiple criteria. Using AND and OR relational operators can sometimes be confusing. Refer to the IF..THEN statement for a full discussion of relational operators.

As a simple example, suppose we wanted to perform an analysis of people over 60 who rate their health as good (1=Good 2=Fair 3=Poor). Both criteria must be true

before we'll select the record (a person must be over 60 **AND** they must be in good health). The command would be:

```
IF AGE > 60 AND HEALTH = 1 THEN SELECT
```

When the AND operator is used, both statements must be true before the record will be selected. The OR operator works differently. When OR is used, the record will be selected if either statement is true. For example, the following command will select people that are Democrats **OR** people that have no political affiliation (D=Democrat R=Republican N=None):

```
IF AFFILIATION = "D" OR AFFILIATION = "N" THEN SELECT
```

A statement can contain as many AND and OR operators as needed. When there are no parenthesis, AND and OR statements will be evaluated from left to right. Parentheses may be used to control the order that the statement will be evaluated. For example, the following command would select all males over 18 and all females over 21.

```
IF (SEX="M" AND AGE>18) OR (SEX="F" AND AGE>21) THEN  
SELECT
```

StatPac contains a special feature that simplifies the syntax of complex OR statements. For example, suppose you want to select respondents from groups 1, 3, 7 and 9. Using OR operators, you would type:

```
IF GROUP=1 OR GROUP=3 OR GROUP=7 OR GROUP=9 THEN  
SELECT
```

The same command in using the simplified OR syntax would be:

```
IF GROUP = "1/3/7/9" THEN SELECT
```

In the first statement, the record will be selected if any part of the statement is true. In the second statement, the OR relational operator is replaced by the slash. When using this method to perform a multiple selection, the codes ("1/3/7/9") must be enclosed in quotes, regardless of the variable type (alpha or numeric). There is no limit on the number of slashes that may be used in the statement to replace OR relational operators. Identical syntax may be used with the REJECT command.

As another example, if some data entry operators entered upper case codes, while other data entry operators entered lower case codes, we would want to select if the code was either upper OR lower case. The following two statements would produce identical results:

```
IF SEX = "M" OR SEX = "m" THEN SELECT
```

IF SEX = "M/m" THEN SELECT

---

## NEW Command

The NEW command is one of several keywords that allows you to create new variables. Its primary application is for creating new alpha-type variables. The advantage of the NEW command over other ways of creating new variables is that it allows you to specify a variable label.

The syntax for the command is as follows:

NEW <(Format)> <"Variable name"> <Variable label>

The format defines the type, field width and decimal formatting (if numeric) for the new variable. It is specified using the same conventions as in the study design except that it is enclosed in parentheses.

The <Format> for creating new alpha variables is:

(Ax)    where x is the field width

The following command would create a new one-column alpha variable named GROUP. The variable label for GROUP would be Group Identification Code.

NEW (A1) "GROUP" Group Identification Code

Quotation marks must be included around the variable name. The variable name itself should be brief (e.g., less than 20 characters in length). The variable label may use a continuation line if the entire label will not fit on the first line. The new variable will be initialized to blanks (i.e., the variable exists, but its value is missing).

Numeric variables may also be created with the NEW command.

The <Format> portion of the syntax for creating new numeric variables is:

(Nx)    where x is the total field width for the variable

or

(Nx.y)    where x is the total field width and y is the  
            number of decimal characters

When working with integer data, the first format is preferable. For example, the following command would create a two-column numeric variable. The variable name is GRAND-TOTAL and its variable label is "The sum of the individual scores". The variable does not have any specified formatting.

NEW (N2) "GRAND-TOTAL" The sum of the individual scores

When creating variables that will contain decimal formatting, the precision for the formatting should be specified. While this is not mandatory, it is highly recommended. For example, let's create a new numeric variable called PROFIT-LOSS. Your study now contains two variables, EXPENDITURES (V1) and INCOME (V2). The first step is to give the new variable a name and/or label. Type:

#### NEW (N10.2) "PROFIT-LOSS" The bottom line

This would create a new numeric variable called PROFIT-LOSS. It will become the third variable in the study. The variable will have a total field width of ten characters (seven to the left of the decimal point, the decimal point itself, and two to the right of the decimal point). The field width refers to the total number of characters reserved for the variable. This includes the space necessary for a decimal point and/or minus sign. In other words, you must have an idea of the magnitude of the new variable before creating it.

After entering the NEW command, the new variable can be referenced by the new variable name (PROFIT-LOSS) or by the new V number (V3). The COMPUTE statement could then be used to calculate a value for the new variable PROFIT-LOSS.

To create several new variables in the same run, simply type several NEW statements on successive lines. Each NEW command will create one new variable.

When running a procedure with the NEW and SAVE commands, the new variable will be created and saved at the end of the codebook and data files. If you attempt to run the same procedure again, StatPac will tell you that you are attempting to create a new variable with the same name as an existing variable (i.e., it was created and saved the first time you ran the procedure. The StatPac.ini file can be edited so StatPac ignores this situation. Set IgnoreDuplicateNewCommand = 1 in the StatPac.ini file to tell StatPac to ignore the NEW command if a variable already exists with that variable name.

When using the NEW command, the variable name is subject to the same restrictions as during the study design.

1. Shorter variable names are preferable.
2. A variable name must be unique from all other variable names and may not be the same as any keyword.
3. The first character of a variable name may not be a number or a space.
4. A variable name may not be the same as a V number. For example, you cannot name a variable "V12".
5. A variable name may not contain a comma or period. The variable name may include a space; however, for the purpose of clarity, we recommend using a dash or underline character instead of a space.
6. A variable name may not be D, E, RECORD, TIME, LO, HI, WITH, BY, THEN, TOTAL or MEAN. These words have special meaning to StatPac.



---

# LET Command

The LET command is used to create a new variable from an existing variable or to assign a new value to an existing variable. The syntax of the command is:

LET <New or existing variable name> = <Existing variable name>

The LET command is often used as a way of making transformations to a data file without destroying the raw data. For example, suppose AGE had been entered in the data file as the respondent's actual age. This would provide excellent descriptive statistics (mean, median, etc.), but it is not conducive to crosstab and banner tables. For these analyses, we want categorical data. If we recode the AGE variable into groups, the original data will be destroyed, since the AGE variable would then contain recoded data (not the raw data). The following procedure overcomes this problem by first using the LET command to create a duplicate copy of a variable, and then recoding the new variable rather than the original variable.

STUDY MARKET

LET AGE\_GROUP = AGE

RECODE AGE\_GROUP (LO-20=1) (21-30=2) (31-40=3) (41-HI=4)

LABELS AGE\_GROUP (1=Under 21) (2=21-30) (3=31-40) (4=Over 40)

SAVE

..

The new variable AGE\_GROUP is created with the LET command. Everything is duplicated except the variable name. AGE\_GROUP will have the same format, variable label, value labels, and data as the original variable AGE. The only thing changed is the variable name. The RECODE command recodes AGE\_GROUP (leaving AGE intact), and the LABELS command assigns new value labels to the recoded data. The SAVE command makes the transformations permanent so subsequent procedures will have access to the new variable AGE\_GROUP as well as the original raw data AGE.

When using the LET command to create a new variable, the new variable name is subject to the same restrictions as during the study design.

The LET command in conjunction with the NEW command can be used to increase the field width of a variable. Suppose you had created a 20 character alpha variable called CITY. After entering several records of data, you decide that you really need 30 characters for the CITY field. You cannot simply change the codebook because you have already entered data, and the existing data needs to be changed in addition to the codebook. The LET command provides an easy solution.

This procedure creates a new variable called NEWCITY and assigns the contents of the existing CITY variable to the new variable for all existing records. After running this procedure, you would add the NEWCITY variable to the data entry form.

STUDY CONTACTS

NEW (A30) "NEWCITY" City

LET NEWCITY=CITY

SAVE

..

When you create a new variable, it is added to the end of the codebook. In the previous example, the NEWCITY variable would become the last variable in the study. The CITY variable would still be in the codebook and on the data entry form. A better solution is to use the WRITE command to eliminate the original CITY variable and replace it with the NEWCITY variable. In this example, CITY was originally variable 25 in the codebook. The WRITE command is used to set NEWCITY as variable 25 and the original CITY variable is omitted.

#### STUDY CONTACTS

NEW (A30) "NEWCITY" City

LET NEWCITY=CITY

WRITE CONTACTS V1-V24 NEWCITY V26-V100

..

The LET command can also be used to convert an alpha (A1) variable to a numeric (N2) variable. This is useful when you have data coded as A, B, C... and you would like to have it coded as 1, 2, 3. You could recode the data (A=1)(B=2)(C=3), save the recoded data, and then manually change the codebook format from A1 to N1, and change the value labels. Alternatively, you can use the NEW and LET commands to do the recode. Suppose APPLE is an (A1) variable coded as A=1 apple, B=2 apples, and C=3 apples. The following two commands would create a new variable (and data) called NUM\_APPLE that was coded as 1=1 apple, 2=2 apples, and 3=3 apples. The new numeric variable must be specified with an N2 format.

NEW (N2) "NUM\_APPLE" Number of Apples

LET NUM\_APPLE=APPLE

---

## STACK Command

A STACK command is used to create new variables that consist of all the possible combinations of categories from two to four other variables. It is especially useful for creating special variables in banner tables.

The syntax for the STACK command is:

**STACK <New variable name> = <Variable list>**

The STACK command may be used only to create a new variable (i.e., it cannot be used to calculate a new value for an existing variable). The format of the new variable will always be alpha, and the field width will be the sum of the individual variables in the variable list. The value labels for the new variable will be automatically created from the combinations of the value labels of the variables in the variable list. The STACK command is temporary and the new variable will only exist in the procedure where the command appears unless the SAVE or WRITE command is used.

As an example, if your study contains a variable called SEX and another variable called AGE, SEX is coded: M=Male and F=Female. AGE is coded: 1=Young,

2=Middle and 3=Old. The following command would create a new variable called SEXAGE that contained six value labels:

**STACK SEXAGE = SEX AGE**

The new SEXAGE variable would be a two-column alpha variable, and its value labels would be:

M1 = Male Young  
M2 = Male Middle  
M3 = Male Old  
F1 = Female Young  
F2 = Female Middle  
F3 = Female Old

The STACK command is a method of adding multiple dimensions to an analysis. If an analysis is performed using the new SEXAGE variable, the results would be based on both the SEX and AGE dimensions.

The STACK command <variable list> may stack up to four variables. For example, a third dimension based on a variable called RACE, could be added with the following command:

**STACK SEXAGERACE = SEX AGE RACE**

The RACE variable (coded: W=White, B=Black and C=Other) would add a third dimension to the DEMOGRAPHIC variable. The new value labels would be:

M1W = Male Young White  
M1B = Male Young Black  
M1O = Male Young Other  
M2W = Male Middle White  
M2B = Male Middle Black  
M2O = Male Middle Other  
M3W = Male Old White  
M3B = Male Old Black  
M3O = Male Old Other  
F1W = Female Young White  
F1B = Female Young Black  
F1O = Female Young Other  
F2W = Female Middle White  
F2B = Female Middle Black  
F2O = Female Middle Other  
F3W = Female Old White  
F3B = Female Old Black  
F3O = Female Old Other

Note that the number of value labels in the new stacked variable is the product of the number of value labels in each of the individual variables in the variable list. Stacking more than two variables in a single command can potentially result in a huge number of value labels.

Also note that the new value labels are created by adding together the value labels that already exist in the study. The value label creation feature of the STACK command works best when each of the value labels for the stacked variables are short.

---

## RECODE Command

The RECODE command is used to recode a variable into groups. It is a data reduction technique used for summarizing data. Both alpha and numeric data may be recoded. The simplest form of the command is:

```
RECODE <Variable or variable list> (<Old value> = <New value>)
```

For example, assume we have a variable called INDICATOR, and we want to change all values of 0 to a value of 5. We would type:

```
RECODE INDICATOR (0=5)
```

The space after RECODE and after the variable name is mandatory. Several variables can be recoded with the same command by specifying a variable list instead of a single variable. The following command would perform the same recode on ten consecutive INDICATOR variables.

```
RECODE INDICATOR1 - INDICATOR10 (0=5)
```

There are several other formats for the RECODE command. One of them allows you to string several recode statements together. For example, let's say you want to change all values of 1 and 2 to a value of 1, all values of 3 and 4 to a value of 2, and all values of 5 to a value of 3. The recode statement would be:

```
RECODE RATING (2=1)(3=2)(4=2)(5=3)
```

Note that it is not necessary to specify (1=1) as part of the recode command. As in the previous example, a variable list could be specified instead of a single variable.

If you prefer, you could reference the variable RATING by its variable number rather than its variable label. Just prefix the variable number with the letter V. For example, if RATING was the third variable in our data file, we could have typed the previous command as:

```
RECODE V3 (2=1)(3=2)(4=2)(5=3)
```

The final format for the RECODE command is used to specify a value range to be recoded instead of an absolute value. The syntax for this type of statement is:

**RECODE <Var. or var. list> (<Low value> - <High value> = <New value>)**

For example, let's say we want to recode all the values from 1 to 20 and give them a new value of 1, and we want to recode all values from 21 to 40 and give them a new value of 2, and finally, all values over 40 should be given a new value of 3. Our RECODE command would be:

**RECODE AGE (1-20=1)(21-40=2)(41-99=3)**

The keywords LO and HI may be included in a RECODE command. LO refers to the lowest value in a data file while HI refers to the highest value in the file. For example, let's say you have a variable called INVENTORY. To change all values from the lowest through 49 to a new value of 0 and also change all values from 50 through the highest to a new value of 1, you would use the following RECODE command:

**RECODE INVENTORY (LO-49=0)(50-HI=1)**

Missing information is stored in StatPac data files as spaces. Spaces may be used in the RECODE command to indicate missing data. For example, let's take the following survey question:

How well do you like our spaghetti?

1. A lot
2. Somewhat
3. Not at all
4. No opinion

To recode all the "no opinion" responses to missing data, type:

**RECODE OPINION (4= )**

Notice that a blank is used as part of the RECODE statement to indicate missing data.

As a similar example, after downloading a twenty-variable data file (DATAFILE) from a mainframe, you discover that all missing data was coded as 99 instead of blanks. Since StatPac recognizes only blanks as missing data, you decide to recode all the variables and save the recoded data. The following commands are used to recode the data file and write a new data file:

**STUDY DATAFILE**

RECODE V1-V20 (99= )

SAVE

..

You may use any of the above formats or combination of formats to create RECODE commands. Many different RECODE commands can be specified in a single procedure. All recodes will be temporary in nature and will be applied to all the tasks in the procedure. A recode can be made permanent by using the SAVE or WRITE commands in the same procedure.

---

## COMPUTE Command

The COMPUTE command is one of the most versatile keywords. It is used to perform algebraic and/or trigonometric functions on numeric variables. The COMPUTE command may be used whenever you want to use arithmetic to transform an existing variable or calculate a new variable. All operations are performed using double precision arithmetic.

The syntax for a compute statement is almost identical to the syntax that the BASIC interpreter uses to evaluate your programs. The format for the compute statement is:

COMPUTE <Variable> = <Equation>

For example, the following equation will add three variables together, calculate the mean average, and replace the contents of variable 9 with the result:

COMPUTE V9 = (V3 + V4 + V5) / 3

Notice that the letter V is used to distinguish a variable number from a constant. You may also use the actual variable names in the equation rather than a V number. For example, the same equation could be written:

COMPUTE DEPT\_AVG = (DEPT\_ONE + DEPT\_TWO +  
DEPT\_THREE) / 3

Compute statements may use five numeric operators and twelve intrinsic functions. They are:

+	addition
-	subtraction
*	multiplication
/	division
^	exponentiation
SQR	square root
LOG	natural log
SIN	sine

COS	cosine
TAN	tangent
ASN	arcsign
ATN	arctangent
ABS	absolute value
EXP	exponent
RND	random (random integer between 1 and the argument)
INT	integer (rounds argument up or down)
FIX	integer (drops decimal portion of the argument)

Equations may also use parentheses to specify the order of computations. If no parentheses are included, computations will be performed in the standard hierarchical order (intrinsic functions / exponentiation / multiplication & division / addition & subtraction). If no hierarchy exists, the equation will be evaluated from left to right. Spaces in an equation will be ignored.

The following are examples of valid equations:

```

COMPUTE V11 = (V22 * 1.3) / (V21 + V16)
COMPUTE V9 = ((V6-V7)*1.42)/9.31
COMPUTE V5 = 0
COMPUTE CIRCUMFERENCE = 3.14159 * DIAMETER
COMPUTE V3 = 3.14159 * V2 ^ 2
COMPUTE V12 = LOG(DOLLARS)
COMPUTE TOTAL-SALES = SQR(V12 - 16.2)
COMPUTE MODIFIED = SIN((ORIGINAL-4.12)+ORIGINAL)
COMPUTE ROUNDED-NUMBER = INT(NUMBER)
COMPUTE TRUNCATED-NUMBER = FIX(NUMBER)

```

The following are invalid equations:

```

COMPUTE V17 = ((V4/V5)    (Mismatched parentheses)
COMPUTE V9 = V6/0         (Division by zero is illegal)
COMPUTE V9 + V3 = V2      (Only one variable allowed to the left
                           of the equals sign)

```

All computations (including the intrinsic functions) are performed using double precision. The result will be rounded to the precision specified by the variable being computed (i.e., the decimal formatting of the variable).

For example, let's say we want to compute an average of three variables. The variable being computed has a format of N5.2. We would write a compute statement to add the three variables and divide by three. If the sum of the three variables is 100, the mean average will be calculated as 33.33333333333333 and the result would be rounded to 33.33.

When no decimal formatting exists for the variable being computed, the result will be expressed with the maximum decimal precision possible. For example, if the format of the computed variable were N7, the previous example would be rounded to 33.3333 (a total field width of seven characters including the decimal point). If the format were N2, the result would be rounded to 33.

If a computed variable becomes too large for the field width of the variable, the precision of the result may be diminished. In the extreme case, where the integer portion of the result would be changed, the result will be set to blanks (missing data). In the previous example, if the format of the computed variable were N1, the result would be stored as missing data because the result (33.333...) could not be expressed using an N1 format. It is, therefore, very important that you have an idea of the magnitude of the number you will be computing.

As another example, suppose we want to add three variables (SCORE1, SCORE2 and SCORE3). The scores are between 0.0 and 99.9 so the variables were originally defined using N4.1 formats. The sum of the variables could exceed 99.9 (four characters), so the new variable that holds the sum (TOTAL\_SCORE) should be created with an N5.1 format.

```
NEW (N5.1) "TOTAL_SCORE" Sum of the three test scores
COMPUTE TOTAL_SCORE = SCORE1 + SCORE2 + SCORE3
```

The COMPUTE statement can also be used to create a new variable directly without first using the NEW command. The syntax to create a new variable is:

```
COMPUTE (<Format>) <New variable name> = <Equation>
```

The only disadvantage of using the COMPUTE command to create a new variable is that a variable label cannot be specified.

The command for the above example is:

```
COMPUTE (N5.1) TOTAL_SCORE = SCORE1 + SCORE2 + SCORE3
```

A new five-character numeric variable would be created and called TOTAL\_SCORE. It will have three digits to the left of the decimal point and one digit to the right. Notice that the format for the new variable is enclosed in parentheses.

Often, equations will become very complex and require many levels of parentheses. While StatPac can handle virtually any level of complexity, it is sometimes easier to break an equation into several smaller equations and store the intermediate results in the variable being computed.

For example, the following complex equation could be broken down into smaller equations:

```
COMPUTE (N6.2) NEWVAR = (V7-V6) + 14.82
```

Could be expanded to:



```

NEW (N6.2) "NEWVAR"
COMPUTE NEWVAR = V7 - V6
COMPUTE NEWVAR = NEWVAR + 14.82

```

Notice that a new variable (NEWVAR) was first created with the NEW command, and then computed as the difference between variables 7 and 6. Finally, it was recomputed to its current value, + 14.82. Because NEWVAR is not already a variable label in the file, it will become the next available variable. In this example, if there were 18 variables already in the file, NEWVAR would become the variable name for variable 19.

Sometimes you will need several COMPUTE statements to accomplish a task. Surprisingly, one of the most difficult formulas you might use is to find the number of days between two dates. Several COMPUTE statements are required. Each date requires three variables in your study (month, day and year). Month and day each require two columns, and year requires four columns. In the following example, the variable names for each date are: MONTH1 DAY1 YEAR1 and MONTH2 DAY2 YEAR2. If the names in your study are different, you must modify this procedure. The DIFF variable contains the number of days between the two dates.

This subroutine can be merged into your procedure file. Note that this example assumes that four digits were used in your study to store the years.

```

COMPUTE (N4) YR1 = YEAR1
COMPUTE (N7) TIME1.0 = 365 * YEAR1 + DAY1
IF MONTH1 > 2 THEN COMPUTE TIME1 = TIME1 -
    INT(MONTH1*.4+2.3)
IF MONTH1 > 2 THEN COMPUTE YR1 = YR1 + 1
COMPUTE TIME1 = INT(TIME1 + (YR1-1) / 4 + MONTH1 * 31)
COMPUTE (N4) YR2 = YEAR2
COMPUTE (N7) TIME2 = 365 * YR2 + DAY2
IF MONTH2 > 2 THEN COMPUTE TIME2 = TIME2 -
    INT(MONTH2*.4+2.3)
IF MONTH2 > 2 THEN COMPUTE YR2 = YR2 + 1
COMPUTE TIME2 = INT(TIME2 + (YR2-1) / 4 + MONTH2 * 31)
COMPUTE (N5) DIFF = ABS(TIME2 - TIME1)

```

The year 2000 is not a problem with the previous procedure because the year was stored using four digits. If you have data that uses only two digits to store the year, then the procedure can be modified to correct the millennium change.

First, determine the oldest year in the data. Suppose in the following example, the oldest value for YEAR1 or YEAR2 is 1921 (stored in the data as 21). By checking each date and comparing it to 21, you can determine if it is a 19xx or 20xx year. Again, the DIFF variable contains the number of days between the two dates.

```

COMPUTE (N2) OFFSET=21
NEW (N4) "YR1"
IF YEAR1 < OFFSET THEN COMPUTE YR1 = YEAR1 + 2000

```

```

ELSE YR1 = YEAR1 + 1900
COMPUTE (N7) TIME1.0 = 365 * YR1 + DAY1
IF MONTH1 > 2 THEN COMPUTE TIME1 = TIME1 -
    INT(MONTH1*.4+2.3)
IF MONTH1 > 2 THEN COMPUTE YR1 = YR1 + 1
COMPUTE TIME1 = INT(TIME1 + (YR1-1) / 4 + MONTH1 * 31)
NEW (N4) "YR2"
IF YEAR2 < OFFSET THEN COMPUTE YR1 = YEAR2 + 2000
ELSE YR1 = YEAR2 + 1900
COMPUTE (N7) TIME2 = 365 * YR2 + DAY2
IF MONTH2 > 2 THEN COMPUTE TIME2 = TIME2 -
    INT(MONTH2*.4+2.3)
IF MONTH2 > 2 THEN COMPUTE YR2 = YR2 + 1
COMPUTE TIME2 = INT(TIME2 + (YR2-1) / 4 + MONTH2 * 31)
COMPUTE (N5) DIFF = ABS(TIME2 - TIME1)

```

There are several types of errors that may occur while using COMPUTE statements (e.g., division by zero or the square root of a negative number). The result of any invalid computation will be set to blanks (missing data).

---

## AVERAGE, COUNT and SUM Commands

The AVERAGE, COUNT and SUM commands are provided to perform calculations in situations in which the COMPUTE command would fail because of missing data. The syntax for all three commands is identical.

The AVERAGE command calculates the mean average of all non-missing values in a list of variables. The syntax of the AVERAGE command is:

```
AVERAGE <Variable> = <Variable list>
```

The COUNT command counts the number of non-missing values in a list of variables. The syntax of the COUNT command is:

```
COUNT <Variable> = <Variable list>
```

The SUM command adds the non-missing values in a list of variables. The syntax of the SUM command is:

```
SUM <variable> = <variable list>
```

The reason for these commands is of the way the COMPUTE command handles missing data. If any of the variables in the COMPUTE statement are missing, the result will be missing. A value calculated by the AVERAGE command is missing only when all values in the list of variables are missing. For example, the following

COMPUTE command would fail if either DEPT\_ONE, DEPT\_TWO, or DEPT\_THREE contained a missing value.

```
COMPUTE DEPT_AVG = (DEPT_ONE + DEPT_TWO +  
DEPT_THREE) / 3
```

This COMPUTE command would store a blank result in DEPT\_AVG for any record in which DEPT\_ONE, DEPT\_TWO, or DEPT\_THREE is missing. Instead, the AVERAGE command could be used to calculate the mean of the non-missing values of the three variables in each record. The commands to accomplish this task would be:

```
STUDY INCOME  
NEW (N9.2) "DEPT_AVG" Average Income of Departments 1-3  
AVERAGE DEPT_AVG = DEPT_ONE DEPT_TWO DEPT_THREE  
SAVE  
..
```

In this example, the DEPT\_AVG variable was created by a NEW command. The format of DEPT\_AVG (N9.2) specifies two places to the right of the decimal point. Therefore, the result of the AVERAGE command will be expressed to two significant decimal places.

The AVERAGE command itself may also be used to create a new variable (making the use of the NEW command unnecessary). The syntax of the command is changed only by the inclusion of the new variable format.

```
AVERAGE (<Format>) <New variable name> = <Variable list>
```

The previous procedure could have been:

```
STUDY INCOME  
AVERAGE (N9.2) DEPT_AVG = DEPT_ONE DEPT_TWO  
DEPT_THREE  
SAVE  
..
```

Note that the format must be enclosed in parentheses. The only disadvantage of using the AVERAGE command to create the new variable is that the new variable will not contain a variable label.

The COUNT command counts the number of non-missing values from a variable list. Its use is identical to the AVERAGE command, except that the result is the number of non-missing values instead of the average of the non-missing values. Like the AVERAGE command, it may also be used to create a new variable. A value calculated by the COUNT command is always an integer between zero and the number of variables specified in the list.

The following procedure creates a new variable DEPT\_COUNT and counts the number of non-missing values in each record. Note that the new variable will always be an integer between 0 and 3, and therefore uses an (N1) format.

```

STUDY INCOME
COUNT (N1) DEPT_COUNT = DEPT_ONE DEPT_TWO DEPT_THREE
SAVE
..

```

The SUM command adds all the non-missing values from a variable list. The result is the sum of these values. Like the AVERAGE and COUNT commands, a new variable may be created. The following procedure would create a new variable called DEPT\_TOTAL that contains the sum of the three departments:

```

STUDY INCOME
SUM (N10) DEPT_TOTAL = DEPT_ONE DEPT_TWO DEPT_THREE
SAVE
..

```

The final example shows all three commands in one procedure:

```

STUDY INCOME
AVERAGE (N9.2) DEPT_AVG = DEPT_ONE DEPT_TWO
DEPT_THREE
COUNT (N1) DEPT_COUNT = DEPT_ONE DEPT_TWO DEPT_THREE
SUM (N10) DEPT_TOTAL = DEPT_ONE DEPT_TWO DEPT_THREE
SAVE
..

```

The result of saving these transformations in a data file could be:

DEPT_1	DEPT_2	DEPT_3	DEPT_AVG	DEPT_COUNT	DEPT_TOTAL
36	43	42	40.33	3	121
49	54		51.50	2	103
27	60	48	45.00	3	135
31			31.00	1	31
				0	
33	44	51	42.67	3	128

---

## IF-THEN ... ELSE Command

The RECODE, COMPUTE and SELECT commands may be modified so that they become conditional. That is, the recode, compute or select will be performed or not performed on a given record depending on whether something else is true or false.

The syntax for the IF-THEN modifier is:

```

IF <Statement> THEN RECODE <Variable> <Recode statement>
IF <Statement> THEN COMPUTE <Variable> = <Equation>
IF <Statement> THEN SELECT

```

Note that the portions to the right of the RECODE and COMPUTE commands have syntax identical to the command when there is no IF-THEN modifier. If the <Statement> portion of the command is true for a given record, THEN the record will be recoded, computed or selected. If the <Statement> portion is false, THEN the record will be skipped.

The following example uses three IF-THEN commands to compute a weighted score based on a group number. Because the SAVE command is used, the weighted score could be referenced in subsequent procedures.

#### STUDY SEGMENT

NEW (N10.4) "WS" Weighted Score

IF GROUP = 1 THEN COMPUTE WS = SCORE \* 0.4172

IF GROUP = 2 THEN COMPUTE WS = SCORE \* 0.8735

IF GROUP = 3 THEN COMPUTE WS = SCORE \* 1.0963

SAVE

..

In the example above, GROUP was a numeric variable and it was not necessary to enclose the values in quotes. If GROUP had been an alpha variable, the procedure would have required quotation marks around the group codes.

#### STUDY SEGMENT

NEW (N10.4) "WS" Weighted Score

IF GROUP = "1" THEN COMPUTE WS = SCORE \* .4172

IF GROUP = "2" THEN COMPUTE WS = SCORE \* .8735

IF GROUP = "3" THEN COMPUTE WS = SCORE \* 1.0963

SAVE

..

When performing an IF-THEN-COMPUTE command, the ELSE keyword may be used to specify an alternate computation if the <statement> portion is false. The following example uses different formulas for males and females to calculate a variable called ADJUSTED-SCORE. The spacing for the continuation lines is for readability only.

#### STUDY SCORES

NEW (N4.2) "ADJUSTED\_SCORE"

IF SEX = "M" THEN COMPUTE ADJUSTED\_SCORE = SCORE \* .59

ELSE

ADJUSTED\_SCORE = SCORE \* 1.37

SAVE

..

The NEW statement in this example could be eliminated by specifying the variable format as part of the COMPUTE statement.

#### STUDY SCORES

IF SEX = "M" THEN COMPUTE (N4.2) ADJUSTED\_SCORE =

```

SCORE * .59 ELSE (N4.2) ADJUSTED_SCORE = SCORE * 1.37
SAVE
..

```

The following is another example of how quotes are used for referencing alpha-type variables. If the format of SEX was A1, (coded M or F), this procedure would select just the males for a descriptive statistics analysis of AGE:

```

STUDY SURVEY
IF SEX = "M" THEN SELECT
DESCRIPTIVE AGE
..

```

When you want to reference missing data, use two quote marks (with or without a space between them). For example, if you want to select all non-missing data from the COMMUNITY variable, either of the following commands would produce the desired results regardless of whether COMMUNITY is alpha or numeric format.

```

IF COMMUNITY <> " " THEN SELECT
IF COMMUNITY = "" THEN REJECT

```

The IF-THEN modifier is often used in conjunction with the COMPUTE command to eliminate the possibility of computational errors. For example, let's say we want to compute the square root of PROFITLOSS. Since we cannot take the square root of a negative number, we only want to perform the transformation if the PROFITLOSS variable is positive. In other words, if PROFITLOSS is greater than zero, take the square root; otherwise, skip it.

In our example, the statement would be:

```

IF PROFITLOSS > 0 THEN COMPUTE PROFITLOSS =
    SQR(PROFITLOSS)

```

Once again, notice that the last part of the statement is identical to the COMPUTE command without the IF-THEN modifier. The only difference is in the IF <Statement> THEN part of the command. The valid relationships supported by StatPac are:

```

=   Equal to
>   Greater than
>=  Greater than or equal to
<   Less than
<=  Less than or equal to
<>  Unequal to
==   Is found within the text string

```

With the exception of the == symbol, all the relational operators are standard algebraic notation. The purpose of the == relational operator is to locate a target string within another string. Its primary use is to search verbatim open-ended

responses for selected words or phrases. For example, suppose V1, V2, and V3 were open-ended multiple response, and respondents' verbatim answers were entered into these three fields. You could use the == relational operator to list comments that mentioned the word "hours". The IF-THEN SELECT line says to search variables one through three for the specified text string (HOURS), and select any record that contains the string. Upper and lower case differences will be ignored when using the == operator.

```
COMPUTE (N5) REC = RECORD
IF V1 - V3 == "HOURS" THEN SELECT
LIST REC V1 - V3
OPTIONS MR=(V1 - V3)
..
```

Often, it may be desirable to search for more than one key word or phrase. The following procedure tells StatPac to search variables one through three for several key words: hours, time, longer, shorter, duration, and length. The LIST command will display the comments that contain any of the search strings.

```
COMPUTE (N5) REC = RECORD
IF V1 - V3 ==
  "HOURS/TIME/LONGER/SHORTER/DURATION/LENGTH/"
  THEN SELECT
LIST REC V1 - V3
OPTIONS MR=(V1 - V3)
..
```

It is important to note that the == relational operator works on the sound of the word, and not the exact spelling. For example, the following procedure might be used to list respondents' positive comments to a multiple response open-ended question.

```
IF ATTITUDE=="Happy/Glad/Satisfied/Pleased" THEN SELECT
LIST ATTITUDE
..
```

The report might look like this. Notice that StatPac found the word "satsfied" even though it was misspelled.

#### Comments

I am very happy with the current program and I cannot think any changes to make.

I am completely satsfied with the new procedures.

I am glad you finally added an evaluation component.

Finally, a program that holds my interest. I am very pleased.

When used in conjunction with other commands, the == relational operator can be used to create a new coded categorical variable from the verbatim text. In a simply example, suppose you asked respondents, "What do you feel is the number one problem in society?" You are especially interested in responses relating to crime,

drugs, violence, and the economy. The following procedure would create a new variable and perform a frequency analysis on it.

```
NEW (N1) "PROBLEM" WHAT IS THE NUMBER ONE PROBLEM IN
      SOCIETY?
LABELS PROBLEM
      (1=CRIME)(2=DRUGS)(3=VIOLENCE)(4=ECONOMY)(5=OTHER)
IF V1 <> " " THEN COMPUTE PROBLEM = 5
IF V1 == "CRIME/CRIMINALS/" THEN COMPUTE PROBLEM = 1
IF V1 == "DRUGS/ALCOHOL/COCAINE/NARCOTICS" THEN
      COMPUTE PROBLEM = 2
IF V1 == "VIOLENCE/VIOLENT/" THEN COMPUTE PROBLEM = 3
IF V1 == "ECONOMY/ECONOMIC/BUDGET/MONEY/" THEN
      COMPUTE PROBLEM = 4
FREQ PROBLEM
..
```

StatPac also supports relational operators AND and OR. They may be used in conjunction with the COMPUTE, RECODE and SELECT keywords (with or without parentheses) to make complex IF-THEN statements. The general syntax is:

```
IF <Statement> AND <Statement> THEN RECODE <Variable> <Recode
      statement>
IF <Statement> OR <Statement> THEN RECODE <Variable> <Recode
      statement>

IF <Statement> AND <Statement> THEN COMPUTE <Variable> =
      <Equation>
IF <Statement> OR <Statement> THEN COMPUTE <Variable> =
      <Equation>

IF <Statement> AND <Statement> THEN SELECT
IF <Statement> OR <Statement> THEN SELECT
```

For example, let's say we want to compute the following equation:

$$\text{NEWVAR} = \text{SQR}(\text{INDEX1}) + \text{SQR}(\text{INDEX2})$$

For this computation to be successful, both INDEX1 and INDEX2 must be greater than zero. (It is not possible to take the square root of a negative number.)

In this example, you could eliminate the possibility of error with the following statement:

```
IF INDEX1 > 0 AND INDEX2 > 0 THEN COMPUTE
      NEWVAR = SQR(INDEX1) + SQR(INDEX2)
```



Because the second line is indented, it is interpreted as a continuation of the previous line. The computation will be performed only if both INDEX1 and INDEX2 are greater than 0.

When using AND relational operators, both statements must be true before the operation (compute, select or recode) will be performed. When using the OR relational operator, if either statement is true, the operation will be performed.

Parentheses may be used in conjunction with AND and OR relational operators to create complex statements. There is no limit on the number of parentheses that may be used in an IF-THEN statement.

Complex weighting schemes can be developed by using combinations of COMPUTE commands. The following procedure creates a weighted subfile where the file is weighted by both age (N2) and sex (A1).

#### STUDY DEMO

```
NEW (N6.4) "CASE_WT"  
IF AGE<21 AND SEX="M" THEN COMPUTE CASE_WT = .9014  
IF AGE<21 AND SEX="F" THEN COMPUTE CASE_WT = 1.2037  
IF (AGE>=21 AND AGE<41) AND SEX="M" THEN COMPUTE  
CASE_WT = .4182  
IF (AGE>=21 AND AGE<41) AND SEX="F" THEN COMPUTE  
CASE_WT = .8109  
IF AGE>=41 AND SEX="M" THEN COMPUTE CASE_WT = .7892  
IF AGE>=41 AND SEX="F" THEN COMPUTE CASE_WT = .8810  
WEIGHT CASE_WT  
WRITE WT_DEMO  
..
```

StatPac contains a special provision that allows the condensation of several OR relational operators. The condensation may pertain to values or variables.

For example, the following command will select a record if the GROUP variable is 1, 4, 5 or 7. If the GROUP variable is any other value, the record will not be selected.

```
IF GROUP=1 OR GROUP=4 OR GROUP=5 OR GROUP=7 THEN  
SELECT
```

This same command could be condensed as:

```
IF GROUP="1/4/5/7" THEN SELECT
```

When using this form of the OR relational operator, the values are separated from each other by slashes. Note that whether you are checking alpha or numeric values, you must enclose the list of values in quotation marks to indicate that there is more than one value to be checked.

Similarly, the following command will select a record if any of the three variables are equal to one:

```
IF RESPONSE_1=1 OR RESPONSE_2=1 OR RESPONSE_3=1 THEN  
  SELECT
```

This same command could be condensed as:

```
IF RESPONSE_1-RESPONSE_3=1 THEN SELECT
```

The variables in the variable list portion of the statement can be listed separately, or as a variable range (or combination of the two). The previous command could also be:

```
IF RESPONSE_1 RESPONSE_2 RESPONSE_3 = 1 THEN SELECT
```

---

## SORT Command

It is sometimes desirable to sort a data file. This is especially true when you will be listing the file and you want the listing to be in some meaningful order. For example, you may want to sort by ZIP code before printing a name and address file.

There are two other situations where a data file must be sorted: 1) before merging two files by a common variable both files must be sorted by the common variable, and 2) before creating an aggregate file, the file must first be sorted by the aggregate variable.

The syntax for the SORT command is straightforward:

```
SORT (Order) <Variable or variable list>
```

The sort order refers to either ascending or descending order and may be specified as (A) or (D). The sort variable(s) may be alpha or numeric format.

An example of the SORT command would be:

```
STUDY NEIGHBOR  
SORT (A) ZIP  
LIST NAME ADDRESS CITY STATE ZIP PHONE  
..
```

The result of this procedure is that the data file will be sorted so the lowest zip code is first and the highest zip code is last. Selected variables from the data will then be listed in sorted order. The SORT command only applies to the procedure in which it appears. Subsequent procedures will use the unsorted data unless the SAVE or WRITE commands were also used in the procedure.

The SORT command is always the last keyword that will be executed in a procedure regardless of where it appears in the procedure (i.e., a file is sorted only after all other transformations have been completed). Therefore, if in the same procedure a variable is both sorted and assigned a new value in a COMPUTE statement, the file

will be sorted according to the newly computed values, regardless of the order of the SORT and COMPUTE lines in the procedure. If you wish to sort by a variable before it is computed, you must sort the variable in a separate procedure before the procedure that computes the variable.

You can use the SORT command to perform a multidimensional sort by specifying a list of variables to be sorted. The variables should be listed in decreasing order of significance; that is, the first variable in the list is the primary sorting variable, the second variable is used only when two values of the first variable are identical, and so on. For example, if you had a file which contained information about people including their last name (LAST\_NAME), first name (FIRST\_NAME), and year of birth (BIRTH\_YEAR), and you wanted to sort the file according to these three variables, you could use the command:

**SORT (A) LAST\_NAME, FIRST\_NAME, BIRTH\_YEAR**

The records in the output file NAMESORT would be ordered alphabetically according to last name. If two or more people in the file had the same last name, their first names would determine who was placed first. If more than one person had the same first and last names, the year of birth would be used to put the records in order. Note that alpha and numeric variables can be combined in the variable list for the SORT command. All variables in the list are sorted in the same order, either ascending or descending.

---

## WEIGHT Command

Often, there are known biases in the sample, and the researcher may want to adjust the sample by weighting cases. This will create a data file that compensates for the bias.

Integer case weighting always produces a fixed output. If a case has a weight of two, it will be duplicated twice in the weighted file. If a case has a weight of three, it will be duplicated three times in the output file.

Non-integer case weighting is based on a probability function and will, therefore, produce different results with each run. If a case has a weight of 2.3, it will be duplicated twice in the weighted file, and there is a 30% chance that it will be duplicated a third time. If a case has a weight of .841, there is an 84.1% chance that it will appear in the output file

This command will allow you to use weights that are already contained in the file, or weights may be assigned to each case depending on the value of another variable in the file.

There are two forms of the command syntax. The first is used when there is a numeric variable in the file that already contains the case weight. The syntax for this form of the WEIGHT command is:

**WEIGHT <Variable>**

For example, if there were a variable called CASEWEIGHT, the syntax would be:

**WEIGHT CASEWEIGHT**

This variable must be numeric and contain the weight of the case. If CASEWEIGHT is missing in any record, the record will be interpreted as if the weight were zero.

If the file does not contain a case weight variable, the other form of the command can be used to assign the weights. The syntax for this form of the command is:

**WEIGHT <Variable> (<Code>=<Weight>)(<Code>=<Weight>)...**

For example, to weight the file based on the respondent's sex (SEX), you would weight each case depending on whether it is coded as M or F. In this example, you want to assign a weight of 1.2 to males and 2.4 to females. The syntax to perform this is:

**WEIGHT SEX (M=1.2)(F=2.4)**

You should enter a weight for each of the codes that exists in the file. The code (on the left of the equals sign) may be alpha or numeric data, while the weight (on the right of the equals sign) must be numeric. If a code exists in the file that is not reflected in the WEIGHT command, it will be assigned a weight of zero.

Notice that the WEIGHT command can produce a file that contains many more records than the original file. You can control the size of the weighted file by adjusting the values of the weights.

If you want the weighted file to contain approximately the same number of records as the input data file, determine the weight for each code by dividing the desired percentage of records containing the code by the observed percentage of records containing the code.

For instance, suppose you have a survey consisting of 150 respondents (100 males and 50 females), and you want to create a weighted data file with 150 records. Also, you want the new file to contain about the same number of males and females (75 males and 75 females). The weight for the male code would be calculated as  $.5/.67$  or  $.75$ . The weight for the female code would be calculated as  $.5/.33$  or  $1.5$ . The command to produce the weighted file would be:

**WEIGHT SEX (M=.75) (F=1.5)**

Note that you can also calculate the weights by dividing the desired number of records by the observed number of records for each code. The weight for males (.75) is equivalent to  $75/100$ , and the weight for females (1.5) is equivalent to  $75/50$ .

Because the non-integer portion of the weight is based on a probability function, the output file will usually not contain the exact number of records as the input file.

Complex weighting schemes can be developed by using combinations of COMPUTE commands. The following procedure creates a weighted subfile where the file is weighted by both race and sex.

**STUDY DEMO**  
**NEW (N6.4) "CASE\_WT"**

```

IF RACE="W" AND SEX="M" THEN COMPUTE CASE_WT = .9014
IF RACE="B" AND SEX="M" THEN COMPUTE CASE_WT = 1.2037
IF RACE="O" AND SEX="M" THEN COMPUTE CASE_WT = .4182
IF RACE="W" AND SEX="F" THEN COMPUTE CASE_WT = .8109
IF RACE="B" AND SEX="F" THEN COMPUTE CASE_WT = .9392
IF RACE="O" AND SEX="F" THEN COMPUTE CASE_WT = .8810
WEIGHT CASE_WT
WRITE WT_DEMO
..

```

It is especially important to include the WRITE command when you use the WEIGHT command. Since StatPac's weighting is based on a probability function, different sets of weighted data will be created each time you run the procedure. Thus, if your intent is the weight the data, and used the weighted data in a series of subsequent analyses, you should use the WRITE command to create a weighted data file that can be used in the subsequent analyses.

```

STUDY MyStudy
WEIGHT CaseWeight
WRITE MyStudy2
..
STUDY MyStudy2
(the rest of the procedures)

```

---

## NORMALIZE Command

A normalized variable is one in which all the values are expressed in terms of standard deviations (Z scores) rather than as the raw data itself. You can normalize any variable or list of variables with the NORMALIZE command.

The formula for a normalized variable is:

$$Z = (X - \bar{X}) / SD$$

where:

- X is the raw data value
- $\bar{X}$  is the mean average
- SD is the standard deviation

A normalized variable will take on positive and negative values. A positive value of Z indicates that the data is above the mean by Z standard deviations, while a negative value indicates that the data is below the mean by Z standard deviations. The format for the NORMALIZE command is straightforward:

```
NORMALIZE <Variable list>
```

For example, to normalize a SALES variable, we would type:

## NORMALIZE SALES

It is also possible to normalize several variables using the same command. Using a similar example, let's say we want to normalize three variables (SALES, ADVERTISING and DIRECT\_MAIL). There are several different ways we could use the NORMALIZE command to normalize all three variables. This involves simply specifying all three variables in the variable list:

```
NORMALIZE V3, V4, V5
NORMALIZE V3-V5
NORMALIZE SALES, ADVERTISING, DIRECT_MAIL
NORMALIZE SALES - DIRECT_MAIL
```

Notice that the only difference between the above commands is the way in which the variable list is specified. The results from each of these would be identical. The variable list may list the variables individually (separated by commas), or by variable range (Low variable - High variable) or by any combination of the two. Either variable numbers or variable names may be used.

Normalized data are non-integer values and contain decimal portions. The number of decimal places is determined by the format of the variable(s) being normalized. Generally, you will not want to normalize the raw data. Instead, create a new variable with the COMPUTE command, save it, and then normalize it in the next procedure. The COMPUTE command will allow you to control the decimal precision of the normalized data.

```
STUDY SALES
COMPUTE (N10.2) "NORM_SALES" = SALES
SAVE
..
NORMALIZE NORM_SALES
SAVE
..
```

The NORMALIZE command may only be used to normalize an existing variable; it may not be used to normalize a new, computed, recoded, or selected variable. Therefore, if you want to preserve the original data, you will need to run two procedures as illustrated in the previous example.

---

## LAG Command

Lagging a variable is often used in simple and multiple regression. When one variable has an effect on another variable, but the effect occurs at a future time, the variable is said to have a lagged effect. A simplified example might be the relationship between our advertising budget and sales. If we double our advertising budget this month, sales will probably increase next month. In other words, advertising budget has a lagged effect on sales. The two variables are related, but one lags behind the other by a specific time period.

The LAG command may be used to lag one or more variables a specified number of time periods. The syntax for the LAG command is:

**LAG (<Number of lags>) <Variable list>**

For example, let's say we wanted to lag variable two by three time periods. The LAG command would be:

**LAG (3) V2**

In essence, when you lag a variable x times, you are pushing the data down x records for that variable (x refers to the number of lags you specify). The consequence of this action is that the data set becomes longer. The following data set illustrates lagging:

<u>Raw data</u>	<u>Lag of one</u>	<u>Lag of two</u>
4	Missing	Missing
9	4	Missing
12	9	4
6	12	9
2	6	12
	2	6
		2

Using our example where ADVERTISING has a lagged effect on SALES, we could look at the two variables before and after ADVERTISING is lagged:

<u>Record #</u>	<u>BEFORE LAG</u>		<u>AFTER LAG</u>	
	<u>Sales</u>	<u>Advertising</u>	<u>Sales</u>	<u>Advertising</u>
1	25	30	25	Missing
2	62	40	62	30
3	80	50	80	40
4	98	63	98	50
5			Missing	63

When you lag a variable in a multiple variable file, the new file will be longer than the original file by the number of lags you specified. The most recent values for the variables that were not lagged will be missing in the new file.

---

## DIFFERENCE Command

Differencing data is a method for removing trend and/or seasonality. Basically, differencing involves subtracting successive observations from each other. The DIFFERENCE command is easy to use and can take differences from data values

one or more time periods apart. To illustrate the concept of differencing, let's look at the following data set. Note that the original data has a well-defined trend (with no irregular values), while the result of the differencing produces a stationary series with no trend.

Raw data	Numbers used to compute a difference	Differenced data
3	6 - 3	3
6	9 - 6	3
9	12 - 9	3
12	15 - 12	3
15	18 - 15	3
18		

Also note that differencing has the effect of reducing the number of records by one. Each time you difference your data, the number of records is reduced.

The format for the DIFFERENCE command is:

**DIFFERENCE (<Periodicity>) <Variable list>**

In the previous example, if the variable were CASH-ON-HAND, the command would be:

**DIFFERENCE (1) CASH-ON-HAND**

If there is more than one variable to difference, simply specify a variable list rather than a single variable name. Use commas or spaces to separate the variable names from one another. The periodicity parameter refers to how many time lags are to be used to calculate the difference. In this example, we are subtracting adjacent values, so the periodicity is one. This is often referred to as a regular or "short" difference because we subtract adjacent values. It has the effect of eliminating trend.

To eliminate seasonality from a data set, do not subtract successive (adjacent) values. Instead, subtract values from the next seasonal period. For example, let's take the following series that has seasonality with a cycle (periodicity) of six periods. That is, the seasonal pattern repeats itself every six periods. In this case, differencing consists of subtracting a value from the corresponding value in the next season. It is known as a seasonal or "long" difference.

Example of seasonal differencing (Periodicity = 6)

Rec. # (Time)	Raw data	Numbers used to compute a difference	Differenced data
1	3	3 - 3 (T7 - T1)	0
2	4	4 - 4 (T8 - T2)	0
3	5	5 - 5 (T9 - T3)	0
4	4	4 - 4 (T10 - T4)	0
5	3	3 - 3 (T11 - T5)	0



6	2	2 - 2 (T12 - T6)	0
7	3	3 - 3 (T13 - T7)	0
8	4	4 - 4 (T14 - T8)	0
9	5	5 - 5 (T15 - T9)	0
10	4	4 - 4 (T16 - T10)	0
11	3	3 - 3 (T17 - T11)	0
12	2	2 - 2 (T18 - T12)	0
13	3	3 - 3 (T19 - T13)	0
14	4	4 - 4 (T20 - T14)	0
15	5	5 - 5 (T21 - T15)	0
16	4	4 - 4 (T22 - T16)	0
17	3	3 - 3 (T23 - T17)	0
18	2	2 - 2 (T24 - T18)	0
19	3		
20	4		
21	5		
22	4		
23	3		
24	2		

Note that the result of seasonal differencing on a series that contains no trend or irregular values produces a perfectly stationary series. Differencing will always result in the loss of data. When you difference for seasonality, the amount of data lost will be equal to one seasonal period. In our example, we lost eight data points because the seasonal period was eight. The command syntax is identical to a "short difference" except that the periodicity parameter is greater than one (i.e., equal to the periodicity).

### DIFFERENCE (8) CASH-ON-HAND

Differencing can, therefore, be used to reduce or eliminate both trend and seasonality, depending on the time lag used for differencing. When the lag is one, the effect will be to eliminate trend. When the time lag is equal to the seasonal period, the effect is to eliminate seasonality.

---

## DUMMY Command

Dummy variables are used in multiple regression to include nominal or ordinal-type data in the regression equation. Normally, only interval or ratio-type data may be used in multiple regression. Dummy variables may only take on values of one or zero. They may be used as independent variables in multiple regression.

Let's say we have a variable that indicates the presence or absence of a credit history. This variable could be coded as a one (meaning there is a credit history), or zero (meaning there is no credit history). This variable could then be included in a multiple regression problem. It is known as a dummy variable because it represents nominal data.

The situation becomes somewhat more complex when there is more than a simple dichotomy (yes/no). Let's take an example where there are several nominal type categories. Extending the previous example, the information for the variable "CREDIT-HISTORY" might have been coded: 1=Excellent history 2=Good history 3=Fair history 4=Poor history. This could be expressed with dummy variables by creating four new variables. The first new variable would be "Excellent history" and would be coded as one (yes) or zero (no). The second new variable would be "Good history" and would also be coded as one or zero, and so on. The coding scheme can be illustrated by the following table:

Raw Data	New var 1 Excellent	New var 2 Good	New var 3 Fair	New var 4 Poor
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	1

Notice that each of the new dummy variables is assigned a value of one or zero depending on the original data value. Obviously, it would be quite time consuming to re-enter four new values (the four new dummy variables) for each record. The DUMMY command will perform this task for you. The syntax for the command is:

**DUMMY <Variable>**

For example, to create dummy variables for the previous example, the command would be:

**DUMMY CREDIT-HISTORY**

This would automatically create four new dummy variables and fill in the values with ones and zeros, depending on the data set. A dummy variable will be created for each unique value that exists in the original data file for that variable. Each dummy variable will be a one-digit numeric value. The result of using the DUMMY command will be the creation of new variables that may then be included in a multiple regression.

The new dummy variable names will be the value labels from the study. For instance, if a variable were coded A=English, B=French and C=Spanish, the DUMMY command would create three new variables named ENGLISH, FRENCH and SPANISH. If the variable does not contain value labels, the variable names for the new dummy variables will be "DUMMY-Vx-y", where x is the variable that was used to create the dummy variables and y is the code from the data that defines the group it came from. For example, if variable 32 were used to create dummy variables, and the data file contained codes A, B and C, the variable names will be "DUMMY-V32-A", "DUMMY-V32-B" and "DUMMY-V32-C".

The DUMMY command will only work if the total number of variables (including the new dummy variables) is not greater than fifty. Do not convert interval or ratio data to dummy variables. This is unnecessary and will usually result in exceeding the fifty variable limit.

Note that a dummy variable will be created for each unique value of the original variable. In order to use these dummy variables in multiple regression, it is necessary

to call one of them the "standard" and not include it in the regression problem. Using all the new dummy variables in the regression equation will result in a singular matrix, and it will not be possible to perform the matrix inversion necessary for regression. Choose one of the dummy variables, call it the "standard", and exclude it from the multiple regression analysis.

The DUMMY command may only be used on a variable that already exists in the data file. A variable that is dummied cannot be computed, recoded, or selected in the same procedure. If you want to apply the DUMMY command to a transformed variable, you must perform the transformation in a separate procedure before the procedure in which the variable is dummied.

---

## RUN Command

The RUN command may be added to the end of any procedure file to enable batch processing of multiple procedure files. It is useful for unattended processing of large jobs, where the size of a procedure file would become excessive. It allows the user to process a series of procedure files in a single batch.

The RUN command may be added as a single line procedure at the end of any procedure file.

The command syntax is:

```
RUN <Procedure file name> <Batch output file name>
```

The <Procedure file name> is the name of the next procedure file to be run. For example, the following procedure file would first run a frequency analysis and a crosstab, and then a new procedure file (PROC2) would be loaded and run.

```
STUDY RESEARCH
FREQ V1
..
CROSSTAB V1 BY V2
..
RUN PROC2
..
```

Multiple procedure files can be run in a batch by adding the RUN command to the end of each procedure file so that at the completion of each procedure file, a new procedure file would be loaded and run. Pre-analysis syntax checking will be performed only on the initial procedure file.

The <Batch output file name> may be optionally specified in the RUN command line to change the batch output file name for the new procedure file. For example, the following command would load the PROC2 procedure file and run it writing the output to FILE2.rtf.

```
RUN PROC2 FILE2
```

If a <Batch output file name> is specified in the RUN command, the new procedure file will write the results to the <Batch output file name>. If page numbering is being used, the new output file will begin with page one, and the table of contents will be appended so that the final table of contents will contain the page numbers for all the procedure files that were run.

---

## REM Command

The purpose of the REM command is to allow you to imbed notes within a procedure file. Comments are especially helpful when reviewing a procedure file that you have not used for a long time. Comment lines will be ignored when performing an analysis. A comment line begins with an apostrophe, or the word REM. There are no restrictions on the text that may be included in a comment line. Comment lines may also use continuation lines. For example, the following procedure contains two comment lines. The second comment also has a continuation line:

```
REM This procedure has two comment lines
STUDY SURVEY12
' This procedure will only use the first 50 records
  for the analysis because the SELECT command is used
SELECT 1-50
FREQUENCIES ATTITUDE
..
```

Comment lines can be useful when debugging a procedure that contains an unknown error. By selectively making each line a comment (adding an apostrophe to the beginning of the line), you can essentially eliminate that line as a possible cause of the error.

Writing complex procedures is often an iterative process. You'll try something, see if it works, and then modify it. Try it again, and modify it... and so on until you get the result you're looking for. Often times, you'll want to save what you've written, and then try something new.

One way to do this is to insert an apostrophe at the beginning of each line you wanted to make a comment because lines beginning with an apostrophe are ignored when processing a procedure file. If you had a procedure file with 3 procedures, and you wanted to completely ignore procedure 2, you'd insert an apostrophe at the beginning of each line in procedure 2, like this:

```
Freq V1
..
' Title (#)
' Desc V2
' ..
Title (#)
Banners V2 By Total V1
..
```

You can also turn entire blocks of text in a procedure file into comment lines. Lines beginning with `/*` can be used to start a comment block, and lines beginning with `*/` can be used to end a comment block.

In the following example, procedure 2 would not be executed because it is enclosed in start and end comment block symbols (`/*` to start and `*/` to end).

```
Freq V1
..
/* Begin ignoring lines from here on
Title (#)
Desc V2
*/ Stop ignoring lines
..
Title (#)
Banners V2 By Total V1
..
```

You can include multiple comment blocks and as many procedures (or individual lines) as you want in each comment block. All lines within a comment block will be ignored while processing a procedure file.

Another way to start and end a comment block is to begin a line with `**` to start and end a comment block. The first line beginning with `**` will start a comment block and the next line beginning with `**` will end the block.

```
Freq V1
..
***** Begin a comment block *****
Title (#)
Desc V2
***** End a comment block *****
..
Title (#)
Banners V2 By Total V1
..
```

If you use the `**` syntax to start and end comment blocks, then be careful because they work in pairs. If you have a start and stop comment and you remove the start comment `**` while inadvertently forgetting to remove the stop comment `**`, then the stop comment `**` would act like a start comment `**` and all lines after it would be ignored.

---

## Reserved Words

In addition to keywords, StatPac recognizes several *reserved words*. Where keywords are always used at the beginning of a line, reserved words are always used somewhere in the middle of a line. Reserved words are specified as a parameter following a keyword or analysis specification command. The reserved words are: RECORD, TIME, TOTAL, MEAN, LO, HI, WITH, BY, THEN, and ELSE. These words have special meaning and should not be used as variable names. These

reserved words may be imbedded in command lines as parameters of keywords, but may not be used as keywords themselves.

---

## Reserved Word RECORD

The word RECORD is a reserved variable name. It is an implicit name built into the command processor and should not be used as one of your variable names. The RECORD variable will always contain the value of the current record number (sequence) that is being read from the data file.

The word RECORD may be used to create a time-series regression. For example, the following command would perform a multiple regression analysis using time (RECORD) and BUDGET as the independent variables and SALES as the dependent variable:

```
REGRESS (2) SALES, RECORD, BUDGET
```

If we had collected yearly data but had not entered the year as a variable, we could use the reserved word RECORD to create the year. The following procedure will create a new subfile (NEWFILE) that contains variables one through nine of the INCOME study, and the new variable YEAR as the tenth variable.

```
STUDY INCOME
NEW (N4) "YEAR"
COMPUTE YEAR.0 = RECORD + 1931   (Our data begins in 1932)
WRITE NEWFILE V1-V9 YEAR
..
```

### **Important User Tip**

There is one caution that should be observed when using RECORD. If the SELECT or SORT commands are used, the record numbers will change so that the record numbers reflect the selected or sorted records. For example, suppose we wanted to list the record numbers of all the cases where AGE is greater than sixty. The following procedure is wrong.

```
STUDY RETIRE
IF AGE > 60 THEN SELECT
LIST RECORD
..
```

Instead, you must first compute the record number, and then reference the computed variable to display the record number. The following procedure is correct.

```
STUDY RETIRE
COMPUTE (N5) REC_NUM = RECORD
IF AGE > 60 THEN SELECT
LIST REC_NUM
..
```

---

## Reserved Word TOTAL

The reserved word TOTAL is used in banners to specify row and column totals in the table. Its use is described in detail under the "Row Totals and Column Totals" section of the BANNERS command documentation.

---

## Reserved Word MEAN

The reserved word MEAN is used in banners to specify row and column mean averages in the table. Its use is described in detail under the "Means and Standard Deviations" section of the BANNERS command documentation.

---

## Reserved Word TIME

The reserved word TIME will be used in a future version of StatPac for Windows.





# Basic Analyses

---

## Analyses Index

These commands may be used in a procedure to set the type of analysis to be performed.

Banners	Breakdown	Correlate	Crosstabs
Descriptives	Frequencies	List	Ttest

These additional commands may be used in a procedure when the Advanced Analyses module has been installed.

ANOVA	Canonical	Cluster	Discriminant
Factor	Logit	Map	PCA
Probit	Regress	Stepwise	

---

## Analyses Overview

There are many different types of analyses that can be performed with StatPac. Most commands are easy to use since much of the required analysis information comes from the default parameter table.

With the exception of the `OPTIONS` command all other analysis commands are mutually exclusive in any given procedure. In other words, a single procedure cannot perform more than one kind of analysis. A procedure file, however, may contain many procedures, each performing a different kind of analysis.

The `OPTIONS` command may be used in any procedure to override the default values in the parameter table. It is used to control printing and analysis parameters.

The analysis commands available in StatPac's basic package are: `LIST`, `FREQUENCIES`, `DESCRIPTIVE`, `BREAKDOWN`, `CROSSTABS`, `BANNERS`, `TTEST`, and `CORRELATE`.

The analysis commands available in the StatPac Advanced Statistics module are: REGRESS, DISCRIMINANT, PCA, FACTOR, ANOVA, CANONICAL, CLUSTER, and MAP.

**Important User Tip**

Any of the analysis commands may be abbreviated by using only the first two characters of the keyword. For example, FREQUENCIES could be abbreviated to FR, FACTOR could be abbreviated as FA, and OPTIONS could be abbreviated as OP.

---

## LIST Command

The LIST command is used to list selected variables in the data file. The command syntax is:

**LIST <Variable list>**

If the LIST command does not specify a variable or variable list, all variables will be listed. When used in this fashion, value labels will be listed instead of the raw data.

For example, let's say you want to print a report consisting of only two columns. The first column is variable 7 (AGE) and the second column is variable 14 (SEX). Either variable numbers or names may be used to specify the variable list. The command line would be entered as either:

**LIST V7 V14**

**LI AGE SEX** (LIST may be abbreviated as LI)

The keyword RECORD may be used as part of the variable list to print the record number as one of the columns. For example, the following parameter line will produce a report consisting of four columns, the first column being the sequence of the case in the data file:

**LIST RECORD V12 V31 V83**

You may specify as many variables to be included in the report that can be accommodated by the pitch and orientation of the output. If too many variables are specified, the output will be truncated. Missing data will be displayed as a series of dashes.

The LIST command is often used as a way to trouble-shoot a procedure that is not working. For example, if the following procedure didn't work properly, we might try the LIST command to figure out what went wrong:

**STUDY EXAMPLE**

**COMPUTE (N4.1) AVG = V1 \* V2 / 2**

**DESCRIPTIVE AVG**

..

We could replace the DESCRIPTIVE command with the LIST command and list the relevant variables. Also note that we added the SELECT command to limit the printout to the first twenty-five records (e.g., we don't need to list the whole file to find out why it is not working).

#### STUDY EXAMPLE

COMPUTE (N4.1) AVG = V1 \* V2 / 2

SELECT 1-25

LIST RECORD V1 V2 AVG

..

### Example of a List Printout

New Product Focus Group Report						
Rec	Taste	Smell	Texture	Appearance	Package	Overall
1	1	2	3	2	1	2
2	1	3	2	3	1	2
3	2	3	4	3	2	3
4	1	2	3	4	3	3
5	3	2	3	2	4	3
6	2	1	2	1	3	2
7	1	3	2	4	3	3
8	3	2	2	2	3	4
9	1	2	3	2	4	3
10	1	4	3	3	1	3
11	2	2	3	2	1	2
12	3	3	2	3	1	2

To list an open-ended variable, simply specify it in the LIST command. The following would list a variable called "Comment". The COMPUTE line is used to calculate a record number so it can be included in the printout. The IF-THEN-SELECT line is used to select only those who made a comment. The OPTIONS line is used insert a blank line between each response.

COMPUTE (N5) REC=RECORD

IF Comment <> " " THEN SELECT

LIST Rec Comment

OPTIONS BL=Y

..

The output might look like this:

### ***Example of a Verbatim Listing***

<b>StatPac For Windows</b>	
Rec	Comments
1	I THINK THAT THE AGENDA SHOULD BE LESS POLITICAL AND GEARED MORE TOWARDS THE AVERAGE PERSON INSTEAD OF JUST A FEW
2	THERE SHOULD BE MORE HANDS ON EXPERIENCE
3	MAKE THE PROGRAMS SHORTER. USE BETTER & MORE HANDOUTS
4	HIRE BETTER PRESENTORS. MORE HANDOUTS.
5	SHORTER HOURS. MORE LOCATIONS.
6	OFFER COURSES MORE OFTEN AND IN MORE LOCATIONS.
7	I AM REALLY TIRED OF PEOPLE TELLING ME WHAT IS IMPORTANT TO LEARN WHEN I FEEL LIKE THIS IS SOMETHING INDIVIDUALIZED FOR EACH PERSON. STOP RAMMING POLITICS DOWN OUR THROATS.

### ***Multiple Response & Combining Variables***

There are two options (MR and CB) to control the way that data gets displayed with the LIST command.

The MR option is used to specify variables you want to be stacked on top of each other in a single column. The specified variables will be listed in a single column rather than using multiple columns on the listing. The variables do not have to be true multiple response variables in the codebook; you may use the MR option for any variables.

The CB option is used to specify variables that you want to combine into a single field instead of being treated as individual fields. For example, if City, State, and Zip were separate variables, you could display them together using the CB option.

Normally, all variables in a listing would appear side-by-side. The MR and CB options are used to create an easier to read format.

For example, suppose variables six, seven, and eight are being used to hold respondents' verbatim answers to a question on a restaurant survey: "What three things could we do to improve your dining experience?" Three A70 variables were used. The following command would produce a listing of the data in a vertical format. Up to three lines in the report would be displayed for each respondent. The SELECT command is used to eliminate subjects who did not answer the question.

```
IF V6 <> " " THEN SELECT
LIST V6-V8
OPTIONS MR=(V6-V8)
```

..

An example of a single record in the printout might look like this:

Faster service.  
Reduce prices.  
Greater selection.

If the CB option were used instead of the MR option, all three responses would be combined into a single field (giving the appearance that the three responses were part of the same sentence or paragraph).

```
IF V6 <> " " THEN SELECT
LIST V6-V8
OPTIONS CB=(V6-V8)
..
```

The CB option formats the printout so each record in the listing will use the number of lines that it needs to show the data. The listing might appear as follows:

Faster service. Reduce prices. Greater selection.

The MR and CB options may be used in conjunction with each other to produce desired outputs. For the next example, assume the following variables:

V1 Name  
V2 Street\_Address  
V3 City  
V4 State  
V5 Zip  
V6 Phone\_Number  
V7 Fax\_Number  
V8 Email\_Address

We might want to stack Name, Address, City, State, and Zip into a single column on the printout. We might also want to stack the Phone and Fax numbers into a single column. In the following procedure, the CB option is used to combine City, State, and Zip into a single field, and the MR option is used to specify which variables should be displayed in a vertical column.

```
LIST V1-V8
OPTIONS CB=(V3-V5) MR=(V1-V5)(V6-V7)
..
```

The output might look like this:

### Example of a Listing Using the MR and CB Options

Listing of Respondents			
Name	Street Address	Phone Number	Email Address
	City State Zip	Fax Number	
David Walbrick	4425 Thomas Ave. S.	612 925-0199	admin@statpac.com
	Minneapolis MN 55410	612 925-0851	
Tom Smith	89512 Lampert Court	301 776-9134	tsmith@hotmail.com
	Baltimore MD 21218	301 776-9135	

### Labeling and Spacing Options

Option	Code	Function
Labeling	LB	Sets the column headings to print the variable label (LB=E), the variable name (LB=N), or the variable number (LB=C). Also, LB=0 suppresses labeling, and LB=X suppresses all labeling and page feeds.
Column Width	CW	Sets the minimum width of the columns (in inches).
Column Spacing	CS	Sets the spacing (in inches) between the columns of the listing.
Maximum Width	MW	Sets the maximum width (in inches) that will be used for long alpha variables and multiple response variables.
Blank Line Between Rows	BL	When BL=Y, a blank line will be printed between each row of the listing. When BL=N, no blank line will be printed.
Maximum Pages	MP	The MP option may be set to the maximum number of pages that will be printed. Its purpose is to prevent an unintentional listing of hundreds or even thousands of pages. If MP=0, then the listing will become as long as necessary to print all the output. If MP is set to any other number, that will become the maximum number of pages that will be printed.

---

## FREQUENCIES Command

A frequency analysis is the simplest of all statistical procedures. It is ideal for data which has been coded into groups or categories. The coding can be either alpha or numeric-type data.

The syntax of the command to run a frequency analysis is:

**FREQUENCIES <Variable list>**

For example, to find the percent of males and females in a sample, you would request a single analysis:

**FR SEX ( FREQUENCIES may be abbreviated as FR)**

Several frequency analyses can be requested with a single command. For example, to get a frequency analysis of SEX (V4), RACE (V5) and INCOME (V6), the request could be specified in several ways:

**FREQUENCIES SEX, RACE, INCOME**

**FREQUENCIES SEX RACE INCOME**

**FREQUENCIES V4 V5 V6**

**FREQUENCIES V4-V6**

Notice that either the variable name or the variable number may be specified as part of the variable list.

A frequency analysis may be run on alpha or numeric-type variables. Missing data will be included in the frequencies only if there is a value label for missing data, (e.g., <BLANK>=No response).

### ***Table Format***

Three types of printout formats are built into the program: expanded, condensed and automatic. The option to control the table format is:

<b>OPTIONS TF=N</b>	(No table will be printed)
<b>OPTIONS TF=A</b>	(Formatting will be automatic)
<b>OPTIONS TF=E</b>	(Formatting will be expanded)
<b>OPTIONS TF=C</b>	(Formatting will be condensed)

Condensed formatting is especially useful when there are many unlabeled values. For example, if one of the variables is ID NUMBER, there are generally no value labels associated with this variable. It is often a good idea to check the data to be sure that no records were inadvertently entered twice (i.e. duplicate ID numbers). A condensed frequencies printout would allow you to quickly determine if any ID NUMBER is specified more than once. An example of condensed formatting might look like this:

### Example of a Compressed Frequencies Printout

StatPac For Windows			
What is your zipcode?			
55303 - 97	55334 - 1	55432 - 47	55421 - 19
55304 - 46	55343 - 15	55445 - 1	55544 - 1
55011 - 13	55070 - 6	55449 - 20	55014 - 26
55112 - 1	55005 - 3	55401 - 2	44533 - 1
55104 - 1			
Valid Cases = 493			
Missing Cases = 54			

Automatic formatting is generally recommended since it minimizes the amount of paper that will be used. If automatic formatting is used and there are more than 50 unlabeled categories (no value labels), the printout will automatically be converted to condensed format. In most cases, this will result in the expanded format. An example of expanded formatting might look like this:

### Example of an Expanded Frequencies Printout

StatPac For Windows		
What is the highest grade or year of school you have completed?	Number	Percent
Never attended or only kindergarden	0	0.0 %
Grades 1-8	4	0.7 %
Grades 9-11	19	3.5 %
Grade 12 or GED (hs grad)	194	35.5 %
1-3 years college or tech. school	180	32.9 %
4 years or more of college (BA	102	18.6 %
Don't know	1	0.2 %
Refused	0	0.0 %
Total	500	91.4 %
Missing Cases = 47		

### Print Zero Values

Sometimes there may be a category listed in the value labels that has no accompanying data. For example, nobody in the sample may be over 40 years old or make over \$30,000 a year. Whether or not you want the label to appear with a count of zero is a matter of preference. If you want the reader of your report to know that a category was available, you'd probably want to print zero values (ZV=Y). If you are interested in saving space, you might want to exclude zero values (ZV=N).

### Sort Type & Sort Order

Frequency analyses are often more meaningful when the output is displayed in sorted order. When working with nominal-type data and few categories, the order in which categories are presented is not very important. (e.g. It really doesn't make much difference whether males or females are listed first.) However, as the number of categories increases, it may be desirable to list those with the highest count first,



followed by those with lower counts. This would be a sort by frequency of response in descending order. It would be requested with the following options:

**OPTIONS ST=F SO=D** (Sort Type by frequency of response)  
(Sort Order is descending)

When data is ordinal, it is more appropriate to present the output in order defined by the categories themselves. Usually this is the same as the alpha or numeric code used to represent a category. For example, take the following two survey questions:

How old are you?	What is your annual income?
A=Under 21	1=Under \$10,000
B=21-30	2=\$10,000-\$20,000
C=31-40	3=\$21,000-\$30,000
D=Over 40	4=Over \$30,000

Both questions are ordinal; the first one is coded alpha and the second is numeric. It would be desirable to have the frequencies printout appear in ascending order by the code (the same way they are listed above). The options statement to do this is:

**OPTIONS ST=C SO=A** (Sort Type is by category code)  
(Sort Order is ascending)

Notice that this type of sort is generally the way the information would be specified in the value labels. If this is the case, sorting by category code will have no effect. Sorting by category codes is useful if you did not enter value labels for the variable.

If no sort type is specified (ST=N), the output will be displayed in the same order as specified by the value labels. If the value labels do not contain all the values in the data file (such as misspelled data), the unlabeled values will appear on the printout in the order that they are encountered in the data file.

Additionally, a digit may be added as a suffix to the SO=A or SO=D. It is used to sort the value labels excluding the last one or more value labels. This is useful when the last value label is an "other" category, and you want to sort the value labels, but still leave the "other" as the last row in the report. For example ST=F SO=D1 would sort the value labels in descending order by frequency, except it would leave the last value label as the last row regardless of its frequency.

### ***Truncate Labels***

Very long value labels may sometimes exceed the space allocated for them in the printout. In those situations, you may set the program to either truncate the value labels (TL=Y excludes the ending portion of the label), or to use multiple lines to print the entire value label (TL=N).

### ***Cumulative Percents***

When the frequency table is printed in expanded format, you may print or exclude cumulative percents with the CP option. This would be specified as:

OPTIONS CP=Y (Turn on cumulative percents)  
OPTIONS CP=N (Turn off cumulative percents)

### **Confidence Intervals**

Confidence intervals for proportions can be requested with the CI option. For example, to request the 95% confidence intervals, you would use the option CI=95, and the 99% confidence intervals could be requested by CI=99. Confidence intervals allow us to estimate the proportions in the population for each of the response categories. If repeated samples are taken from the population, we would expect the category proportions to fall within the confidence intervals. When confidence intervals are requested, cumulative percents will not be printed regardless of the setting of the CP option.

Confidence intervals are calculated by first computing the estimated standard error of the proportion, and then using the t distribution to find the actual interval. Note that the finite population correction factor  $(1-n/N)$  is used to adjust the standard error if the sample represents a large proportion (say greater than ten percent) of the population. When the sample is large, use the FP options to specify the population size (i.e., FP=x, where x is the size of the population). If the FP option is not specified, no correction will be applied.

### **Example of an Confidence Intervals Around A Percent**

StatPac For Windows			
Are you currently married?	Number	Percent	95% CI
Yes	328	60.0 %	± 4.2 %
No	171	31.3 %	± 4.0 %
Total	499	91.2 %	
Missing Cases = 48			

### **Critical T Probability**

After performing a frequency analysis, researchers are often interested in determining if there is a significant difference between the various categories. The Chi-square statistic is often used to determine if the observed frequencies markedly differ from the expected frequencies. The problem with the Chi-square statistic is that it does not isolate the significant differences (i.e., it only tells whether or not one exists). StatPac uses a t-test to compare all possible pairs of categories to determine where the actual differences lie.

The CT option may be set between 0 and 1. When CT=0, no t-tests will be performed or printed. If CT=1, the t-statistic and probability will be printed for all possible pairs of categories. A typical setting for the critical t probability is 5% (CT=.05). In this case, StatPac will print the t-statistic and two-tailed probability for all pairs of categories that have a probability of  $p=.05$  or less. StatPac uses the following formula to calculate the t-statistic:

## Example of a Critical T Probability Analysis

StatPac For Windows		
Who do you plan to vote for in the next election?	Number	Percent
Jesse Ventura	27	33.8 %
Norm Colman	18	22.5 %
Skip Humphrey	17	21.3 %
Other	12	15.0 %
Undecided	6	7.5 %
Total	80	100.0 %
Missing Cases = 0		
T-tests between group percents - (Values of p are for a two-tailed test.)		
Note: Statistics are printed only if p is less than or equal to .05		
$\chi^2(78)=2.494, p=.015$	4 = Jesse Ventura 1 = Other	
$\chi^2(78)=4.005, p=.000$	4 = Jesse Ventura 5 = Undecided	
$\chi^2(78)=2.547, p=.013$	3 = Norm Colman 5 = Undecided	
$\chi^2(78)=2.373, p=.020$	2 = Skip Humphrey 5 = Undecided	

## Percentage Base

The percentage base on a frequency analysis can either be the number of respondents (N) or the total number of responses. If PB=N, the denominator for calculating percentages will be the number of respondents. If PB=R, the denominator will be the total number of responses for all individuals.

## Multiple Response

Surveys often include questions in which the respondent is asked to make more than one response to a single question. An example of the kind of question that is appropriate for multiple variable response is:

1. Which of the following services did you use?

(Check all that apply)

- ☐ Counseling
- ☐ Job placement
- ☐ Remedial reading
- ☐ Remedial math
- ☐ Resume writing

The multiple response frequency analysis is used to summarize these kinds of items. When designing a study that includes this type of question, each choice is considered as a separate variable. The value labels need to be specified only for the first variable, but it is fine if they are specified for all the multiple response variables.

V1 "Services\_1" Services Used  
1=Counseling  
2=Job placement  
3=Remedial reading  
4=Remedial math  
5=Resume writing

V2 "Services\_2" Services Used

V3 "Services\_3" Services Used

V4 "Services\_4" Services Used

V5 "Services\_5" Services Used

The syntax for the multiple response frequency analysis is:

```
FREQUENCIES <Variable list>  
OPTIONS MR=Y
```

In this example, all the variables in the variable list will be treated as one multiple response variable. Another way to use the MR option is to re-specify the variable numbers (not the variable names) that should be grouped.

```
FREQUENCIES <Variable list>  
OPTIONS MR=(<Variable list>)
```

Note that the parentheses are required around the variable list in the options line. In the above example, the commands would be:

```
FREQUENCIES V1-V5  
OPTIONS MR=(V1-V5)
```

The output will contain the counts and percents for each of the response values. That is, how many times code 1 (counseling) was chosen for any variable, how many times code 2 (job placement) was chosen for any variable, etc. In other words, it will print the total number of times that each response was recorded for variables 1, 2, 3, 4 and 5 combined.

The options line may be used to specify several multiple response analyses by using additional sets of parentheses in the MR option. The following commands would perform three different tasks (each one being a multiple response analysis on a new set of variables).

```
FREQUENCIES V1-V20
```

OPTIONS MR=(V1-V10)(V11-V15)(V16-V20)

Multiple response may also be used when the questionnaire limits the choices to less than the number of possible responses. For example, the following question asks for two responses from the same value labels list:

17 & 18. Write the numbers of your two favorite foods from the list below.

- 1 = Hotdogs
- 2 = Hamburgers
- 3 = Fish
- 4 = Roast Beef
- 5 = Chicken
- 6 = Salad

Notice that there are two variables (17 & 18) that hold the information for this question. Both variables use the same value labels and the responses to both variables are weighted equally (i.e. the first one is not more important than the second). Multiple response assumes that all variables to be analyzed have the same value labels. In this example, the command would be:

FREQUENCIES V17 V18

OPTIONS MR=(V17 V18)

### ***Example of a Multiple Response Frequency Analysis***

StatPac For Windows			
<u>Radio Stations Listened To During The Rating Period</u>			
	Number	Percent	Cumulative
WABC	8	38.1 %	38.1 %
WKNT	8	38.1 %	76.2 %
WPIE	7	33.3 %	109.5 %
WWDR	4	19.0 %	128.6 %
WXYZ	6	28.6 %	157.1 %
Total	33		
Number of Cases = 21			
Number of Responses = 33			
Average Number Of Responses Per Case = 1.6			
Number Of Cases With At Least One Response = 19			
Response Percent = 90.5 %			

If you have imported the data from an external source, the multiple response data might not be in StatPac format. Several other commercial software packages use dichotomous multiple response, where the data for all multiple response variables is either one or blank..

The MX=Code option can be used in conjunction with the MR option to tell StatPac that the data for the variables are dichotomous. The "Code" is the single character value that indicates the item is selected. In the above example, the data was coded as ones and blanks, so MX would be set to 1. If the data had been coded as Y and N, then MX would be set to Y.

```
Frequencies V1-V4
Options MR=Y MX=1
..
```

Using this method does not actually change the data file. StatPac just reads the data differently for the frequencies procedure. An exclamation mark cannot be used to permanently set the MX option. It must be explicitly specified in each procedure where you want to use it. If you want to permanently change the data, use the Conversion Utility program.

### ***Category Creation***

The actual categories in the frequency analysis can be created either from the study design value labels (CC=L) or from the data itself (CC=D). When the categories are created from the labels, the value labels themselves will be used to create the categories for the analysis, and data that does not match up with a value label code will be counted as missing. That is, misspunched data will be counted as missing. When categories are created from the data, all data will be considered valid whether or not there is a value label for it.

### ***One Analysis***

The one-analysis option allows you to print frequency analyses for several variables on one page. This option is especially useful for management reporting when the information needs to be condensed and concise.

All the variables specified with the OA option must have the same value labels. An example might be a series of Yes/No questions or Likert scale items. The important point is that each variable has exactly the same value labels as the other variables. For example, suppose that variables 21-30 are ten items asking the respondents to rate the item as low, medium or high. The following commands would produce a one page summary of all ten items:

```
FREQUENCIES V21-V30
OPTIONS OA=Y
```

The one analysis option is limited by the number of characters that can be printed on a line (i.e., by the pitch and carriage width of the printer). If there are too many different value labels, they will not be able to fit on one line and the analysis will be skipped. If this should happen, try rerunning the analysis using a compressed pitch. As a general rule, each value label will require ten spaces on the output.

### ***Example of a One-Analysis Printout***

## New Product Focus Group Report

### General Attribute Ratings

(N=31)

	Excellent 1	Good 2	Fair 3	Poor 4	Total
How would you rate the taste?	8 27.6%	7 24.1%	11 37.9%	3 10.3%	29 100.0%
How would you rate the smell?	4 14.3%	9 32.1%	11 39.3%	4 14.3%	28 100.0%
How would you rate the texture?	1 3.6%	12 42.9%	14 50.0%	1 3.6%	28 100.0%
How would you rate the appearance?	4 14.3%	10 35.7%	10 35.7%	4 14.3%	28 100.0%
How would you rate the package design?	6 21.4%	10 35.7%	8 28.6%	4 14.3%	28 100.0%
What is your overall rating for this product?	2 7.1%	10 35.7%	12 42.9%	4 14.3%	28 100.0%

The OA option is used in frequency analyses to summarize the frequencies of several variables that all contain the same value labels. Note the difference between the OA and MR options. With the multiple response option (MR), the items are treated as if they are a single variable. The one analysis option (OA), however, treats each item as a separate analysis. The results, however, will be summarized on one page.

The keyword RECORD in the variable list creates a blank line in the report. For example, the following will print three rows, a blank row, and then two more rows

```
Freq V1 V3 V7 Record V9 V12
Options OA=Y
```

..

When the MR option is used in conjunction with the OA option, the variables in the MR options list will be treated as multiple response variables. This makes it easy to create nets in a frequencies with the OA=Y option.

For example, if V1-V20 are the twenty variables, we could add a net by first creating a duplicate copy of V1 with a new name, and then including the MR option to combine the variables to make the net. The net will be the sum of the counts of the individual variables that make up the MR variable list.

```
STUDY Yourstudy
NEW (N1) "Grand-Total"
COMPUTE Grand-Total = V1
LABELS Grand-Total (1=Agree)(2=Neutral)(3=Disagree)
FREQ Grand-Total V2-V20 V1-V20
OPTIONS OA=Y MR=(Grand-Total V2-V20)
..
```

The results might look like this:

	Agree	Neutral	Disagree
Grand-Total	-----	-----	-----
Variable 1	-----	-----	-----
Variable 2	-----	-----	-----
Variable 3	-----	-----	-----
etc.			

The following is another example shows how you can use MR option in conjunction with the OA option to create complex nets. It also shows how the reserved word "RECORD" can be used to create blank lines in the report.

Suppose we are conducting of survey of government policies. We have nine "Agree/Disagree" items coded as 1=Agree and 2=Disagree. The first three items deal with "Social Policy"; the next three items with "Foreign Policy"; and the last three items with "Fiscal Policy". We would like to produce a report that looks something like this:

#### Peoples Attitudes Towards Government Policies

(N=x)	<u>Agree</u>	<u>Disagree</u>
OVERALL	----	----
SOCIAL POLICY	----	----
Item 1	----	----
Item 2	----	----
Item 3	----	----
FOREIGN POLICY	----	----
Item 4	----	----
Item 5	----	----
Item 6	----	----
FISCAL POLICY	----	----
Item 7	----	----
Item 8	----	----
Item 9	----	----

There are four different nets in this report. The OVERALL net includes all variables. The SOCIAL POLICY net includes the first three items, the FOREIGN POLICY net the next three items, and the FISCAL POLICY net the last three items. For this example Items 1-9 are stored in variables 1 to 9.

The spacing (indentation) in this example is used only to make the procedure easier to understand. It is not necessary to use the this type of spacing in your procedures.

**STUDY Yourstudy**



```

HEADING Peoples Attitudes Towards Government Policies
COMPUTE (N1) OVERALL=V1
COMPUTE (N1) SOCIAL POLICY=V1
COMPUTE (N1) FOREIGN POLICY=V4
COMPUTE (N1) FISCAL POLICY=V7
LABELS OVERALL
      SOCIAL POLICY
      FOREIGN POLICY
      FISCAL POLICY
      (1=Agree)(2=Disagree)
FREQ OVERALL V2-V9          Produces overall net
RECORD                     Produces a blank line
SOCIAL POLICY V2-V3        Produces social policy net
V1-V3                      Produces 3 social policy variables
RECORD                     Produces a blank line
FOREIGN POLICY V5-V6       Produces foreign policy net
V4-V6                      Produces 3 foreign policy variables
RECORD                     Produces a blank line
FISCAL POLICY V8-V9        Produces fiscal policy net
V7-V9                      Produces 3 fiscal policy variables
OPTIONS SV=N OA=Y
      MR=(OVERALL V2-V9)
      (SOCIAL POLICY V2-V3)
      (FOREIGN POLICY V5-V6)
      (FISCAL POLICY V8-V9)
..

```

### ***Special Value Label HIDE***

When performing a frequencies with the OA option, it is often desirable to only display some of the response categories. Recoding undesirable categories to missing is one method to exclude it from the table. This will eliminate the column from the table and from any calculations of percentages on the table.

For example, assume the following counts for V1:

1	2	3		
Agree	Neutral	Disagree	No Response	Total N
30	20	40	10	100

If PB=N, (denominator equals number of respondents), the percents will be:

Agree	Neutral	Disagree
30%	20%	40%

If PB=R, (denominator equals number of responses), the percents will be:

Agree	Neutral	Disagree
30/90=33%	20/90=22%	40/90=44%

We could use the following RECODE command to eliminate the "Neutral" category from the table:

```
RECODE V1 (2= )
```

If PB=N, the percents will still be based on a denominator of 100. If however, PB=R, then the percents will be based on a denominator of 70 (30+40):

Agree	Disagree
30/70=43%	40/70=57%

The special value label "HIDE" may be used to suppress printing of a value label without reducing the denominator for the percents calculations. The following LABELS command could be used to eliminate the "Neutral" category from the table, while still including the "Neutral" count in the denominator:

```
LABELS V1 (1=Agree)(2=Hide)(3=Disagree)
```

Any row or column that has a value label of "HIDE" will not be printed, but it will be included in the percent calculations when PB=R. Note that the percentages are based on the counts for all value labels (including the "Neutral" category), even though all the value labels are not displayed in the table.

Agree	Disagree
30/90=33%	40/90=44%

If you only wanted the "Agree's" to show in the table, you could use the following statements in the procedure. The percentages in the table would still be based on 90:

```
LABELS V1 (1=Agree)(2=Hide)(3=Hide)
OPTIONS PB=R
```

### ***Print Format***

The results from the one analysis option may be printed as row percents (PF=R), as counts (PF=N), or both (PF=NR). When row percents are requested, the denominator used to calculate the percents will be the number of non-missing responses for that particular item. That is, when there is missing data, the number of valid responses to a particular question may be different than the number of valid responses for any of the other questions.

### ***Print Total***

The PT option may be used in conjunction with the OA (one-analysis) option to print the total N for each variable. When there is considerable missing data, this option is highly recommended since each of the variables may be using a different N (number of valid responses). For example, the following commands would produce a one-page report summarizing variables 21 to 30. An additional column will be included on the output that lists the number of valid cases for each of the variables.

```
FREQUENCIES V21-V30
OPTIONS OA=Y PT=Y
```

### ***Sort Variables***

When performing a frequency analysis with the OA=Y option, you can sort the variables by the contents of the first column of the results. The SV (sort variables) option may be set to "N" for no sort, "A" to sort in ascending order, or "D" to sort in descending order. When no sort is specified, the variables will be listed in the order that they appear in the analysis command variable list. The SV option is applicable only when the OA=Y option is specified. However, if the MR option is also specified, the SV option should be set to N.

Additionally, a digit may be added as a suffix to the SV=A or SV=D. It is used to sort the variables excluding the last one or more variables when the OA=Y option is specified. This is useful when the last variable is an "other" variable, and you want to sort the variables, but still leave the "other" as the last variable. For example SV=D1 would sort the variables in descending order, except it would leave the last variable as the last row regardless of its value.

### ***Supplemental Heading***

The supplemental heading will only be printed when the OA=Y option is specified. It is a line of text that will appear before the first row of the table. The supplemental heading may contain any text and should be enclosed in quotes. When the pounds symbol is used in the supplemental heading, it will be printed as the number of cases. The SH option is usually used to indicate who is included in the table. The following is an example of a supplemental heading:

```
OPTIONS SH="TOTAL RESPONDENTS = #"
```

### ***Minimum Denominator***

Percentages can be misleading if they are based on a small denominator. The MD option may be used to suppress the printing of percentages that are based on a small denominator. The MD option sets the minimum denominator that StatPac will use for calculating percents. For example, if MD=5, StatPac will calculate percentages if the denominator is greater than or equal to 5. If a denominator were less than 5, StatPac would print dashes instead of the percent. Valid values for MD are between 0 and 100. If MD=0, all percentages will be printed.

### ***Print Mean***

The mean average is generally not calculated for a frequency because it involves the assumption of interval data. However, there are some situations where you may want to display the mean as part of a frequency analysis. The ME option may be used to

request the mean (and standard deviation). When ME=N, no mean will be printed. If ME=Y, the mean will be printed, and if ME=S, both the mean and standard deviation will be printed. When used in conjunction with the OA=Y option, a separate mean will be printed for each variable.

### ***Mean Position***

When the ME option is used with the OA=Y option, the means (and standard deviations) can be printed as the first or last column. If MP=F, the means will be printed as the first column, and when MP=L, they will be printed in the last column. When means are printed in the first column (MP=F), and the SV option is used to sort the variables, they will be sorted by the means instead of the percents.

### ***Labeling and Spacing Options***

<b>Option</b>	<b>Code</b>	<b>Function</b>
Labeling	LB	Sets the column headings to print the variable label (LB=E), the variable name (LB=N), or the variable number (LB=C).
Labeling Width	LW	Set the maximum width (in inches) for the variable labels on the stub when OA=Y.
Truncate Labels	TL	Sets long value labels to be truncated when TL=Y.
Exact Width	EW	When OA=Y and EW=Y, the labeling width for the stub will be exactly what is specified with the LW option. When EW=N, the width of the stub will self-adjust based on the length of the stub labels.
Column Width	CW	Sets the minimum width of the columns (in inches) when OA=Y.
Column Spacing	CS	Sets the spacing (in inches) between the columns of the listing when OA=Y.
Extra Spacing	ES	When ES=Y, a blank line will be printed below the table headings. When ES=N, no blank line will be printed.
Blank Line Between Rows	BL	Sets the number of blank lines between rows when OA=Y.
Print Percent Symbol	PP	Sets whether percentage symbols will be shown.
Decimal Places	DP	Sets the number of decimal digits that will be shown.
Print Codes	PC	Sets whether the code (to the left of the equals symbol) will be shown with the value labels.

### ***Open-Ended Response Coding***

It is often useful to code open-ended responses into response categories in order to perform a frequency analysis or crosstabs. The FREQUENCIES command with the OE option allows you to examine and code open-ended alpha variables.

Open-ended response coding is requested by performing a frequency analysis on the variables containing the verbatim text and setting the option OE=Y. Do not use an IF-THEN SELECT command in the same procedure.

```
FREQ Comment
OPTIONS OE=Y
```

```
..
```

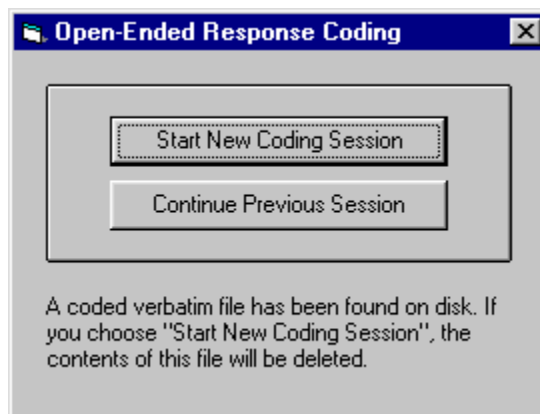
When a verbatim response is held in more than one variable, it is not necessary to specify the MR (multiple response) option. All variables listed in the Frequencies command will be considered to be part of the same verbatim comment. The following procedure says to begin an open-ended response coding session on variables one through three. Note that the text for all three variables will be displayed at the same time during the coding session.

FREQ V1 - V3  
OPTIONS OE=Y

..

Each time you complete a coding session, StatPac will create a new study and data file called STATPAC-VERBATIM. The STATPAC-VERBATIM study contains the coded verbatim data, and the frequency analysis will be performed on this file (i.e., the coded data). Your original study and data file are not affected by the coding process. All of the coded information is stored in the STATPAC-VERBATIM file. In order to use this coded information in future analyses, the STATPAC-VERBATIM files must be merged with your study and data files using the MERGE command.

Occasionally, you may start a coding session, and for one reason or another, not be able to finish. You can quit the current coding session and continue at a future time. To continue with a previously unfinished coding session (i.e., one that you started coding before, but did not finish), simply run the procedure again. StatPac will detect the existence of a partially completed STATPAC-VERBATIM file, and ask if you want to continue with the previous verbatim coding, or delete the existing verbatim coding and begin a new coding session.



Click on the Continue Previous Session button to continue with the previous coding, or the Start New Coding Session button to delete the existing STATPAC-VERBATIM files. If your intent is to continue with a previous session, you can bypass this question by changing the OE option from OE=Y to OE=C (continuation).

### **Verbatim Blaster**

StatPac for Windows has the Verbatim Blaster module built in and it will automatically pre-code all open-ended responses.

Verbatim Blaster processes open-ended responses in two steps. The first step is called *pre-coding*, and the second step is called *final coding*. You may modify or interact with the coding process at either or both steps.

The pre-coding step will be performed first. Verbatim Blaster will read the file and count all the unique words that occur in the text. It attempts to combine variations of the same word into a single root word. For example, it would attempt to combine singulars and plurals, different tenses, prefixes and suffixes into a single root word. The pre-coding is not perfect, but it will catch nearly all variations on each root word.

The result of the pre-coding will be to present you with a list of root words along with the number and percent of respondents who used each word. The list will be initially sorted in descending order by frequency of occurrence in the text. Thus, the words at the top of the list appeared most often in the text. If a respondent used the same word more than once, it will only be counted as one occurrence in calculating the frequencies.

The pre-coding screen appears:

**Open-Ended Response Coding**

☒ Step 1 - Examine Words  
☐ Step 2 - Select Words for Categories  
☐ Step 3 - Final Coding

Type of Sort: ☒ Frequency ☐ Alphabetic  
 Minimum Percent:    
 (27 words)

Records = 200  
 Responses = 134  
 Total Words = 1411  
 Words Per Response = 10.5

Select words to join or to add to the exclusion list

Words	Count	Percent
showcase	43	32.1%
time	36	26.9%
workshop	32	23.9%
hall	31	23.1%
exhibit	28	20.9%
hotel	25	18.7%
artist	18	13.4%
present	17	12.7%
conference	15	11.2%
food	14	10.4%
hour	13	9.7%
day	11	8.2%
meeting	11	8.2%
booth	10	7.5%
people	10	7.5%
room	10	7.5%

Context: 1 200

Agent Manager & State groups in smaller meetings. More exhibit time. Presenter education of manager/agent roles. problems, etc.

## Step 1 - Examine Words

The most important task in Step 1 (pre-coding) is to familiarize yourself with the basic content of the verbatim comments.

When no words are selected (highlighted) in the word list, the Previous and Next buttons will show the previous and next records.

A more important feature is the ability to examine the context in which specific words are used. First select the word or words you are interested in exploring by clicking on those words in the word list. Then the Previous and Next buttons can be used to find the previous and next comments that used any of the selected words. The selected word(s) will be shown in red to draw your eyes to that portion of the comment.

Clicking on a word that is not already selected will select the word. Clicking on a word that is already selected will deselect the word.

### ***Join Word Variations***

During pre-coding, one of the major functions of Verbatim Blaster is to combine all the variations of each root word. It is generally a good idea to review the words and to combine any variations that Verbatim Blaster may have missed. It will usually not miss any, but it's still a good idea to check.

In the Type of Sort window, click on Alphabetic. Then use scroll bar to scroll through the list. It will be easy to spot variations of the same root word that were not combined since they will appear next to each other in the alphabetically sorted list.

If you should find two variations of the same root word, you will want to join them together so Verbatim Blaster treats them as a single unique word. First select the words to be joined by clicking once on each word. The selected words will be highlighted. You may join more than two words at once by clicking on each word. Then click the Join button. The words will be joined and the count and percent will be modified to reflect the new values. The important thing is to join words that are variations of the same root word or words that have the same meaning.

You may join words during pre-coding or final coding. However, joining word variations is easiest during the pre-coding process because of the ability to perform an alphabetical sort.

When you have finished joining words, click on Frequency in the Type of Sort window to re-sort the list in descending order by frequency of occurrence. During the final coding process, it is usually most convenient to have the most common responses appear near the top of the response category list.

### ***Delete and Exclude Words***

There are many words that add little meaning to a sentence. Words like *of, the, for, by, this*, and hundreds of others don't add much to the meaning of a respondent's statement. These words could be excluded from a sentence without substantially distracting from its meaning.

Modifiers are words that are used to describe the quantity or magnitude of the word or phrase that follows the modifier. They are usually adverbs. Examples are *usually, mostly*, and *greatly*. Modifiers do add meaning to a sentence, but they are generally not helpful in determining the major topic of a sentence.

Verbatim Blaster maintains a list of exclusion/modifier words in a file called EXCLUDE.TXT. This file is an ASCII text file and may be edited with any word processor or text editor. It contains an alphabetical listing of words that do not help identify the primary topic of a sentence.

The first thing Verbatim Blaster does when you ask it to analyze a text file is to eliminate the exclusion/modifier words. It doesn't really eliminate the words; it just pretends they're not there.

The exclusion/modifier list distributed with Verbatim Blaster is fairly complete. However, specific applications may require that you add new words to this list. You can use any text editor to modify the EXCLUDE.TXT file. Words added with a text editor will be sorted into alphabetic order the next time that Verbatim Blaster is run.

You can also add exclusion/modifier words to this file during pre-coding. If you see a word that doesn't add substantial meaning to sentence, add it to the exclusion/modifier list by clicking on the word to select it, and then clicking on the Exclude button. The word(s) will be added to the EXCLUDE.TXT file and will be excluded from all future open-ended coding session.

Deleting words is different than excluding words. Deleting a word with the delete button will delete the word from the current session only. Future coding sessions with other verbatim text would show the word. To delete a word, click on the word to highlight it and click the Delete button.

### ***Set the Minimum Percent to Display***

A typical text file might contain 400 to 500 unique root words (even after combining all the variations of the same words and eliminating the exclusion words). Usually, we wouldn't want to look at a list of this size. Instead, we're most often interested in responses that were made by more than one respondent. Verbatim Blaster lets you adjust the size of the word list by specifying a *minimum percent*. This is the minimum proportion of respondents that used a word. For example, if you set the minimum percent equal to five, Verbatim Blaster will display words that were used by at least five percent of the respondents. Words that were used by less than five percent of the respondents would be hidden from your view.

At any time during pre-coding, you may change the minimum percent. When the minimum percent is set to zero, all words will be displayed. To change the minimum percent, simply type the new minimum percent and press enter.

### ***Step 2 - Select Words for Categories***

The purpose of pre-coding is to identify the important words that are mentioned by respondents. "Important" is, of course, a subjective decision. The minimum percent feature will narrow the number of words to a manageable list. Step 2 is to select those words that seem to hit upon the key concepts in respondents' answers. Sometimes these will be easy to identify, and other times they won't. If the survey question was extremely specific, it will probably be easy to identify the key concept words, and if the survey question was quite general, it might be extremely difficult to identify the key concepts.

"Select Words for Categories" refers to identifying the key concept words that will be carried forward to the final coding process. To select a word, or to deselect a word that has already been selected, click on it in the word list.

If you select any words, only those words will be carried forward to the final coding process, and words not selected will be excluded. If you do not select any words, all displayed words will be carried forward to final coding.

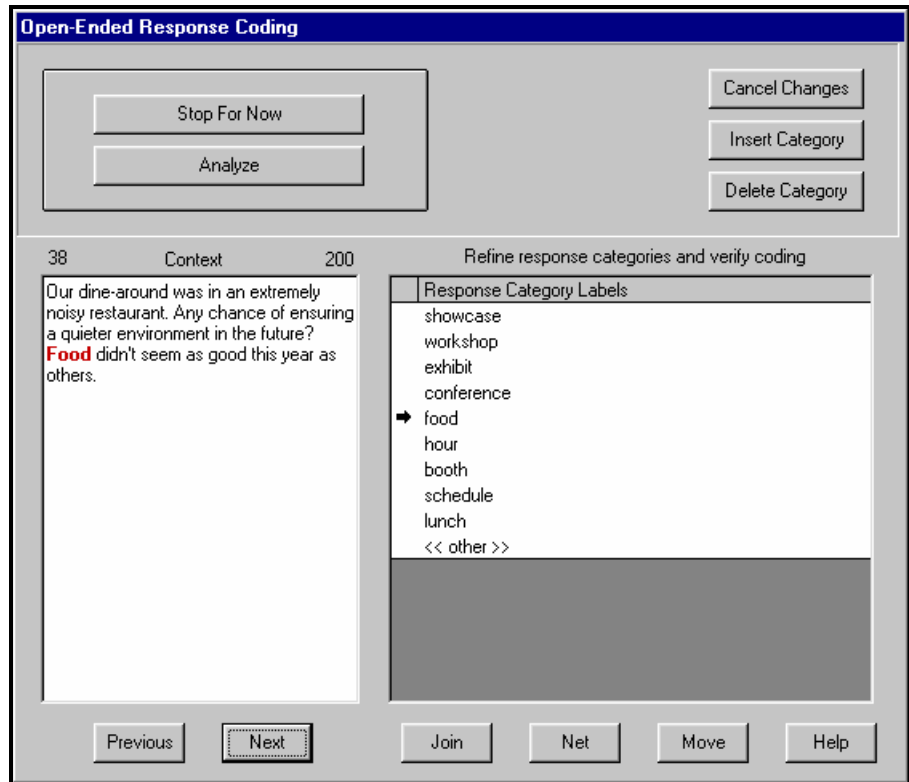
### ***Step 3 - Final Coding***

The final coding process is where you refine the coding and the response category labels. The words selected in the pre-coding process provide the foundation for the final response categories. These are the labels you give to the key concepts. The initial response categories will be the words that were selected during pre-coding.

The final coding process involves reviewing the actual open-ended responses for each respondent, and using your understanding of the comment(s) to refine the response category labels. A response category label can be changed at any time. To change the text in a response category label, double click on the label.

The *text window* (respondent's verbatim text) will appear at the left of the screen, and the *response categories window* will appear on the right. There will be an arrow to the left of all response categories that were mentioned by the respondent, and the key words in the text will be highlighted.





### Select and Deselect a Response Category

Use the mouse to select and deselect a response category. When a response category is selected, an arrow will appear to the left of that category. This means that the current respondent made a comment related to that response category. If a response category is not selected, there will be no arrow. Clicking to the left of the response category label (in the small area reserved for the arrows) will select or deselect that response category.

### Change Records

A *record* is the same as a respondent. Thus, when we say *changing records*, it simply means displaying a different respondent's answer. There are two ways to change records.

The first way is used to show a specific desired record. Click on the record number shown on the top left of the Context window. After clicking on the record number, change it to the desired record and press enter.

The second way is to use the Previous and Next buttons to change to the previous and next records. When no response category labels are highlighted, the Previous and Next buttons will advance to the previous or next record numbers. When one or more response category labels are highlighted, the previous or next record that has been pre-coded into that category will be displayed. The current record number will be displayed above and to the left side of the context windows.

One method of performing the final coding would involve repeatedly click the Next button to review the coding beginning with the first respondent and going to the last respondent. Verbatim Blaster will skip over respondent's who did not make a comment.

Another method of pre-coding would be to examine the comments for each response category. First, highlight the response category (or categories) you want to search

for. Then click the Next button to search for the next record that contains a reference to that response category. Each time you click the Next button, the next record with a reference to that response category will be displayed. When the last record in the text file is reached, the search will be stopped.

The search feature provides a quick way to gain a better understanding of a particular response category. It lets you scan all comments related to a specific response category. While using the search feature, the search will be limited to the response category currently being searched. This makes it extremely easy to scan the relevant text. Scanning a particular response category will give you a better understanding of the comments coded into that category.

### ***Change a Response Category Label***

Sometimes, it might be necessary to change or delete an existing response category, or to create a new category. The response category labels can be changed at any time by simply typing the new text. Double-click on the response category label you want to change and then you will be able to edit the category label.

### ***Delete and Create New Response Categories***

To create a new response category, highlight a response category and click the Create Category button. This will insert a blank line in the response categories so you can type the new response category label on that line. If you create a new response category using this method, you will need to go through each record and decide whether or not that record falls into the newly created response category. Verbatim Blaster does not automatically code responses based on the words you type.

To delete a response category, highlight it and click the Delete Category button. The response category will be immediately eliminated. There is no automatic "undelete", so be careful. You might use Delete Category button to eliminate a response category you consider to be unimportant.

### ***Join Two Response Categories***

Sometimes you will want to combine response categories that you initially thought were different. You may join response categories (at any time) into a single category. Click the Join button. Then drag and drop one of the categories onto the other category. The category you dragged will be deleted and all the responses that were initially assigned to that category will be reassigned to the category you drop it on. To drag a category, move the mouse pointer over that category. Press and hold the left mouse button. While still holding the mouse button, move the mouse so the category outline is over the category to be joined and then release the mouse button. Note that once you use the join category feature, there is no way to return to the unjoined version. There is no automatic "unjoin", so be careful.

### ***Create a Net Response Category***

Creating a *net* category is a useful method of aggregating responses. It is similar to joining response categories except that the secondary response category is not removed as a unique entry in the response category list.

The most common use of a net category is to summarize a group of related response categories without affecting the existing categories. For example, suppose you were evaluating respondents preferences for a new food and there were response categories of red, green, and blue. You might want to create a net category called color. You could use the Join button to join the three categories, but you would then be unable to break down the respondents by their individual color choices. Creating a net category is the solution to the problem.

To create a net category, first create a new blank line for the Net category. Highlight a category and click on the Insert Category button to open up a blank line in the response category list. This blank line will become the net category. Next click the Net button to begin net creation. Finally, drag the category you want to net and drop it on the blank line. Additional response categories can now be added to the new net category. Click Net again, and drag another category to the new Net category. Response categories may be included in a net one at a time using this method.

If you make a mistake while creating or adding to a net, click on the Cancel button to cancel the process. If you inadvertently add a wrong variable to a net, delete the net with the Delete Category button and recreate the net.

### ***Change the Order of the Response Categories***

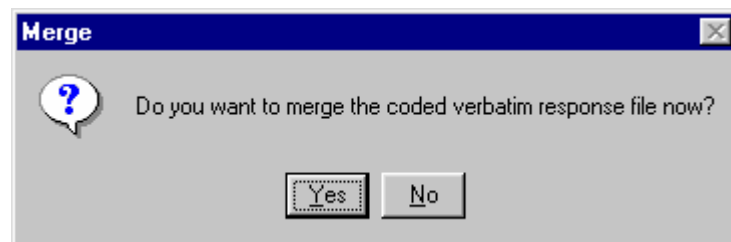
It is sometimes desirable to rearrange the order of the response categories. To move a response category, click the Move button. Then select a category, drag it to a new position, and drop it in the new position in the response category labels list

### ***Finish the Coding Process***

If you wish to exit the open-ended response coding program before finishing the coding process, make a note of the current record number and click the Stop For Now button. To continue where you left off at a future time, run the same procedure again, and select Continue Previous Session. Then click on the current record number on the top left of the Context window, type the record number where you left off, and press enter.

To finish the coding process, and run the frequency analysis on the coded data, click the Analyze button. The frequency analysis will be performed on the coded data.

After viewing the results of the analysis, StatPac will ask if you want to merge the coded verbatim responses. For example, you may want to use the coded verbatim information in other analyses (e.g. crosstabs of the verbatim responses with other variables). Since StatPac can only analyze one study at a time, you should merge the coded responses (i.e., the STATPAC-VERBATIM file) with your original study and



After running the merge procedure, your original file will contain all the original data (including the original verbatim comments) and the new coded verbatim comments. The coded comments (from the STATAC-VERBATIM file) will be added to the end of your original variables, so running the procedure will increase the number of variables in your original study.

### ***Produce a List of Verbatim Comments for Each Response Category***

After merging a coded verbatim file, you can merge the STATPAC-VERBATIM procedure file into your existing procedure file. The STATPAC-VERBATIM procedure file is created automatically when you do a merge. It contains a series of procedures to print a listing of the verbatim responses that were coded into each response category. To merge the file, position the cursor at the beginning of the line

following two dots. Then select File, Merge, and select STATPAC-VERBATIM as the file to merge.

---

## CROSSTABS Command

Crosstabs is one of the easiest ways to look at the relationship between two variables, and one of the most popular ways of examining categorical data.

The syntax for the crosstabs analysis is:

**CROSSTABS <Variable list> BY <Variable list>**

For example, let's look at how people's expectations for learning (EXPECTATION) are related to their satisfaction with a lecture (SATISFACTION). The command to request this crosstab analysis is:

**CR EXPECTATION BY SATISFACTION**

(CROSSTABS may be abbreviated CR)

The results will be printed in the form of a two-dimensional matrix. The first variable (EXPECTATION) will be printed on the y axis, while the second variable (SATISFACTION) will be printed on the x axis. The keyword BY is a mandatory part of the statement.

If several different crosstabs are desired, request them by specifying a variable list instead of an individual variable. For example, you might be interested in both SATISFACTION with the lecture and the amount of actual LEARNING that occurred. The command to run this analysis would be:

**CROSSTABS EXPECTATION BY SATISFACTION, LEARNING**

The matrix size that the crosstabs program can accommodate depends on the available RAM. The variables themselves may be alpha or numeric. StatPac will not print a row or column when total count for that row or column is zero. Missing data (blanks) will be excluded from the analysis unless there is a value label for blank data (e.g., BLANK=Missing data).

Three-way crosstabs may be requested by the following command:

**CROSSTABS <Var. list> BY <Var. list> BY <Var. list>**

A three-way crosstab is essentially a series of two-way crosstabs controlled for a third variable. That is, the two-way crosstabs are performed on subsets of the data as defined by the third variable. For example, consider the following crosstabs command:

**CROSSTABS EXPECTATION BY SATISFACTION BY SEX**

This command will produce two different crosstab tables, one for males and the other for females. The same results could be obtained by executing the following two procedures:

```
IF SEX="M" THEN SELECT
CROSSTABS EXPECTATION BY SATISFACTION
..
IF SEX="F" THEN SELECT
CROSSTABS EXPECTATION BY SATISFACTION
..
```

### ***Count/Percent & Observed/Expected Tables***

There are two common ways to print crosstabs. One is number, row percent, column and total percent. The second is observed, expected, observed minus expected, and the cells contribution to the total chi-square. Both of these tables may be printed or excluded using Y or N options. To print both tables, use the following options:

```
OPTIONS CP=Y OE=Y
```

The chi-square is an important statistic; it is used to test whether two variables are independent of each other. In other words, do the observed frequencies in the cells deviate markedly from the frequencies we would expect if the two variables were not related to each other?

A large chi-square statistic indicates that the observed frequencies differ significantly from the expected frequencies. A crosstab with  $r$  rows and  $c$  columns is said to have  $(r-1)$  times  $(c-1)$  degrees of freedom.

Using the chi-square distribution and its associated degrees of freedom, you can calculate the probability that the differences between the observed and expected frequencies occurred by chance. Generally, a probability of .05 or less is considered to be a significant difference; this probability is termed "probability of chance" in the output.

When a crosstab contains many cells with counts less than five, the probability of chance for the chi-square statistic can be inaccurate. Therefore, the user should consider grouping some rows and/or columns if many cells have expected values less than five.

The second way of printing crosstabs (observed/expected table) is useful in explaining the significance of the chi-square statistic. The cells with high values in the "contribution to the chi-square" are the ones that "contribute" the most to the significance of the chi-square. This is useful in the discussion of the results of a study as there are often only a few cells which deviate from independency.

## Example of a Count/Percent Table

StatPac For Windows					
<u>The Relationship Between Lecture Expectation and Satisfaction</u>					
<u>Crosstabs: How would you rate your expectation for this seminar? - By -</u> <u>How would you rate your satisfaction with this seminar?</u>					
Number Row % Col % Total %	How would you rate your satisfaction with this seminar?				Totals
	Bad 1	Fair 2	Good 3	Excellent 4	
A=Low	10 27.0% 38.8% 10.5%	15 40.5% 53.6% 15.8%	9 24.3% 33.3% 9.5%	3 8.1% 13.0% 3.2%	37 38.9%
B=Medi..	2 13.3% 11.8% 2.1%	4 26.7% 14.3% 4.2%	7 46.7% 25.9% 7.4%	2 13.3% 8.7% 2.1%	15 15.8%
C=High	5 11.6% 29.4% 5.3%	9 20.9% 32.1% 9.5%	11 25.6% 40.7% 11.6%	18 41.9% 78.3% 18.9%	43 45.3%
Totals	17 17.9%	28 29.5%	27 28.4%	23 24.2%	95 100.0%

## Print Format

Each cell of the crosstabs table may contain up to four numbers. Their meanings are labeled in the upper left corner of the table. You may choose to print or suppress any of these numbers by using the PF option. The parameters for this option are:

- N Number or observed frequency
- R Row percent or expected frequency
- C Column percent or observed minus expected
- T Total percent or contribution to chi-square

One or more parameters may be used with the PF option. These should not be separated from each other. For example, if you want to print the number and total percent, use the following option:

**OPTIONS PF=NT**

If a table is too large to fit on one page, it will be split to use as many pages as necessary. The actual number of columns that can fit on a page is determined by the pitch and carriage width of your printer.

## Category Creation

The actual categories (rows and columns) in the crosstab analysis can be created either from the study design value labels (CC=L) or from the data itself (CC=D).

When the categories are created from the labels, the value labels themselves will be used to create the categories for the analysis, and data that does not match up with a value label code will be counted as missing. That is, mispunched data will be counted as missing. When categories are created from the data, all data will be considered valid whether or not there is a value label for it.

### **Sort Codes**

The actual labeling for the x and y axes are taken from the value labels. In most circumstances, the order that you entered the value labels (during the study design) reflects the order in which you want the value labels to be listed. You can override the order of the value labels in the study design by using the option (SC=Y). The value labels will then be displayed in ascending alphabetical or numeric order. This feature is especially useful when the study design itself does not contain any value labels. If this option is not used (i.e., SC=N), the order of the value labels on the printout will reflect the order in which values are encountered in the data file.

### **Statistics**

When the statistics option is specified, several other statistics will be calculated and printed.

### **Example of a Statistics Printout**

Chi-Square = 17.8	Valid Cases = 95
Degrees Of Freedom = 6	Missing Cases = 2
Probability Of Chance = .007	Response Rate = 97.9%
Cramer's V = .3	Somer's D = .4 (X Indep)
Contingency Coeff. = .4	Somer's D = .3 (Y Indep)
Tau-A = .2	Somer's D = .3 (Symm)
Tau-C = .3	Gamma = .5
Entropy = 3.3	Lambda = .2
Distribution Index = .9	

A discussion of each statistic follows:

### **Phi**

The Phi statistic is calculated and printed for two-by-two tables. It may be interpreted as a measure of the strength of the relationship between two variables. When there is no relationship, Phi is zero. When there is a perfect positive relationship, Phi is one. When there is a perfect negative relationship, Phi is minus one.

When comparing one crosstab table to another, Phi is preferable to the chi-square because it corrects for the fact that the chi-square statistic is directly proportional to the number of cases. In other words, Phi could be used to compare two crosstabs with unequal N's.

### **Cramer's V**

If Phi is calculated for tables larger than two-by-two, there is no upper limit to its value. Therefore, the Phi statistic is not printed for tables greater than two-by-two. Instead, Cramer's V is printed. Cramer's V adjusts the Phi for the number of rows and columns so that its maximum value is also one. It may be interpreted exactly like the Phi (e.g., a large Cramer's V indicates a high degree of association between the two variables).

## ***Contingency Coefficient***

The contingency coefficient is another measure of association based on the chi-square statistic. It may be calculated for any size of table; however, its maximum value will vary depending on the number of rows or columns. Therefore, the contingency coefficient should only be used to compare tables with the same numbers of rows and columns.

## ***Kendall's Tau Statistics***

Kendall's tau statistics are used to measure the correlation between two sets of rankings. It is the number of concordant pairs of observations minus the number of discordant pairs adjusted so it has a range of minus one to plus one. There are three different methods for standardizing tau (tau-a, tau-b and tau-c). Note that tau-b is only calculated for square tables.

## ***Gamma***

Gamma is similar to the tau statistics except that it may be interpreted directly as the difference in probability of like rather than unlike orders for the two variables when they are chosen at random. Gamma has a value of plus one when all the data is in the diagonal that runs from the upper-left corner to the lower-right corner of the table. It has a value of minus one when all the data is concentrated in the upper-right to lower-left diagonal.

## ***Cohen's Kappa***

Cohen's Kappa is another measure of the degree to which the data falls on the main diagonal. It is only calculated for square tables.

## ***Somers' d***

Somers' d is a measure of association for ordered contingency tables when there is a dependent and independent variable. It may be interpreted in the same fashion as a regression coefficient.

## ***Odds ratio***

The odds ratio is calculated for two-by-two tables. Its value may vary between zero and infinity. A value greater than one indicates a positive relationship while a value near zero represents a negative relationship. A value of one indicates statistical independence. Note that this is different than most measures of association.

## ***Yule's Q and Yules Y***

Yule's Q is a function of the odds ratio. Like the odds ratio, its value will vary between zero and one; unlike the odds ratio, a value of zero indicates statistical independence, while values of minus one and one represent perfect negative and positive relationships. It will be calculated for two-by-two tables.

## ***Entropy***

Entropy is a measure of disorder; that is, the extent to which the data is randomly distributed in a contingency table. The greater the disorder, the greater the entropy statistic. It is useful for comparing different crosstab tables with each other. A low entropy (near zero) indicates that the data tends to be clustered in only a few of the possible categories. A high entropy indicates that the data is evenly distributed among all the possible categories.



## Yate's Correction

If degrees of freedom equals one (i.e., when the crosstabs produces a two-by-two table), the chi-square statistic can have the Yate's correction applied and be printed as the "Corrected chi-square". The option YA=Y will enable Yate's correction for two-by-two tables, while YA=N will disable it.

## Residual Analysis

Residual analysis is one method used for identifying the categories responsible for a significant chi-square statistic. This involves calculating the standardized residual for each cell and adjusting it for its variance. The normal distribution is used to find the probability of the adjusted residual using a two-tailed test of significance. A significant adjusted residual indicates that the cell made a significant contribution to the chi-square statistic.

The residual analysis may be turned on or off with the option RA=Y and RA=N, respectively. A sample printout of a residual analysis would look like this:

### Example of a Residual Analysis Printout

StatPac For Windows				
<u>The Relationship Between Lecture Expectation and Satisfaction</u>				
<u>Residual Analysis: How would you rate your expectation for this seminar? - By -</u> <u>How would you rate your satisfaction with this seminar?</u>				
	How would you rate your satisfaction with this seminar?			
Std. Residual				
Variance				
Adj. Residual				
Probability	Bad	Fair	Good	Excellent
	1	2	3	4
A=Low	1.313	1.240	-0.467	-1.991
	0.501	0.431	0.437	0.463
	1.855	1.890	-0.707	-2.926
	0.064	0.039	0.480	0.003
B=Medi..	-0.418	-0.200	1.326	-0.856
	0.691	0.594	0.603	0.638
	-0.302	-0.260	1.707	-1.072
	0.615	0.795	0.088	0.284
C=High	-0.971	-1.032	-0.349	2.352
	0.449	0.386	0.392	0.415
	-1.449	-1.661	-0.558	3.652
	0.147	0.097	0.577	0.000

## Interaction Analysis

While many of the statistics indicate whether or not two variables are related, Goodman's interaction analysis is a method of finding out if the magnitude of the relationship is caused more by one part of the table than another. Its purpose is to evaluate all possible combinations of two-by-two tables for interaction effects.

The interaction is defined as the natural log of the odds ratio. The purpose of the log function is to take into account the possibility of a curvilinear relationship. The standard error of the interaction is calculated as well as the standardized interaction. The standardized interaction is used to calculate a two-tailed probability using a normal distribution.

The interaction analysis may be requested with the IA=Y option. A sample printout would look like this:

### ***Example of an Interaction Analysis Printout***

<b>StatPac For Windows</b>					
<b><u>The Relationship Between Lecture Expectation and Satisfaction</u></b>					
Absolute and Standardized Values of Interaction Effects (DF = 18)					
2x2 Table Specification		Interaction	Standard Error	Standardized Interaction	Two-Tailed Probability
Rows	With Columns				
A & B	With 1 & 2	0.3	1.0	0.3	.764
A & B	With 1 & 3	1.4	0.9	1.5	.142
A & B	With 1 & 4	1.2	1.2	1.0	.315
A & B	With 2 & 3	1.1	0.8	1.4	.156
A & B	With 2 & 4	0.9	1.1	0.9	.393
A & B	With 3 & 4	-0.2	1.0	-0.1	.882
A & C	With 1 & 2	0.2	0.7	0.3	.792
A & C	With 1 & 3	0.9	0.7	1.3	.207
A & C	With 1 & 4	2.5	0.8	3.0	.003
A & C	With 2 & 3	0.7	0.6	1.2	.248
A & C	With 2 & 4	2.3	0.8	3.1	.002
A & C	With 3 & 4	1.6	0.8	2.1	.038
B & C	With 1 & 2	-0.1	1.0	-0.1	.919
B & C	With 1 & 3	-0.5	1.0	-0.5	.631
B & C	With 1 & 4	1.3	1.1	1.1	.253
B & C	With 2 & 3	-0.4	0.8	-0.5	.642
B & C	With 2 & 4	1.4	1.0	1.4	.148
B & C	With 3 & 4	1.7	0.9	2.0	.049

### ***Equiweighting***

Equiweighting is a technique to eliminate distortions from most measures of association caused by column marginal disparities. You should use Equiweighting whenever there is a dependent/ independent variable relationship (implying causality) and the column totals differ markedly for each of the categories. Note that Equiweighting only applies to the observed/expected table and the statistics that are printed with the table. After Equiweighting, cell frequencies will no longer be integer values. Equiweighting may be requested with the EQ=Y option.

### *Labeling and Spacing Options*

Option	Code	Function
Labeling	LB	Sets the column headings to print the variable label (LB=E), the variable name (LB=N), or the variable number (LB=C).
Labeling Width	LW	Set the maximum width (in inches) for the variable labels on the stub.
Exact Width	EW	When EW=Y, the labeling width for the stub will be exactly what is specified with the LW option. When EW=N, the width of the stub will self-adjust based on the length of the stub labels.
Column Width	CW	Sets the minimum width of the columns (in inches).
Column Spacing	CS	Sets the spacing (in inches) between the columns.
Key	KY	Sets whether the top left corner of the banner will show a legend of the cell contents.
Print Percent Symbol	PP	Sets whether percentage symbols will be shown.
Decimal Places	DP	Sets the number of decimal digits that will be shown.
Print Codes	PC	Sets whether the code (to the left of the equals symbol) will be shown with the value labels.
Label Justification	LJ	Sets the justification for the banner variable label.
Label Underline	LU	Sets whether the banner variable label will be underlined.
Column Justification	CJ	Sets the justification for the banner value label columns.
Column Underline	CU	Sets whether the banner value label columns will be underlined.
Bottom Justify	BJ	Sets whether the banner labels will be bottom justified.

---

## BANNERS Command

Banner crosstabs are often used in marketing research when it is important to display several crosstab tables as part of the same printout. It is similar to the crosstabs program except that multiple variables may be specified for the x and/or y axis. The variables across the top of the page are called the banners and the variables down the side of the page are called the stub. It has an advantage over regular crosstabs in that there is much more control over the appearance of the output. The disadvantage is that not all the statistical measures of association are available with banners.

The syntax for the command to run banners is:

**BANNERS** <Stub variable list> **BY** <Banner variable list>

The keyword **BY** is a mandatory part of the command syntax. The first variable list (called the stub) will be displayed down the side of the page, and the second variable

list (called the banner) will be displayed across the top of the page. The maximum table size is 250 rows and 60 columns. The variables may be alpha or numeric.

To print banners with variables 1 & 2 on the y axis and variables 3 to 7 on the x axis, enter the command:

```
BA V1 V2 BY V3 - V7 (BANNERS may be abbreviated as BA)
```

If a table is too large to fit on one page, it will be split over as many pages as necessary.

Vertical bars may be inserted into the banner variable list to force a page break at that position between banner variables. The following would create at least two pages. Variables 3 and 4 would be the banner for the first page and variables 5, 6, and 7 would be the banner for the second page. The vertical bar is inserted in the list as if it were a variable (separated by commas or spaces from the other variables).

```
BA V6 BY V3 V4 | V5 V6 V7
```

```
..
```

Multiple vertical bars can be used and each | will create a page break at that position in the table.

The actual number of columns that can fit on one page is determined by the many parameters including font size (pitch), column width, spacing between banner columns, spacing between banner variables, and zoom factor. Most users run their banner tables in the landscape mode (OR=L) in order to increase the number of columns that can fit on a page.

Value labels appearing in the banner heading will be split into multiple lines as necessary to fit in the banner column widths. The actual positions of the word splits can be controlled by inserting a vertical bar into the value labels at the locations where you want the words to split. For example, a value label of "Government" might break into two lines in the banner heading. You could force the break to occur between the n and m by changing the value label to Govern|ment or Govern-|ment. Use a LABELS statement with a ! suffix to change the value labels.

```
LABELS Organization (1=Govern|ment)(2=Private)(3=Non-|Profit)!  
BANNERS V9 BY TOTAL Organization
```

```
..
```

### ***Type of Data***

Two types of banner tables can be printed. The most common type is the count/percent table. The data for the rows and columns is categorical (nominal or ordinal). Each row and column in the table represents a category. Set TY=C to select the count/percent table. Unlike crosstabs, **the banners program defines its row and columns from the value labels in the study design.** That is, the program uses the value labels to create the banner rows and columns. Any data value not having a matching value label in the study design will be counted as missing; therefore, set up the study design labels to reflect the headings and labeling for the banners.

The second type of table is when the stub variables are interval or ratio data. There aren't any defined categories for the rows. Instead of counts and percents, we would

want to see means and standard deviations in the table. Set TY=P to indicate that the stub variables are parametric. The table will show means and standard deviations instead of counts and percents.

### ***Print Format***

Each cell of the banners table may contain up to four numbers. To print or suppress any of these numbers, use the PF option.

The parameters for this option are:

- N    Number
- R    Row percent
- C    Column percent
- T    Total percent

One or more parameters may be used with the PF option. These should not be separated from each other. For example, to print the number and total percent, use the following option:

**OPTIONS PF=NT**

## Example of a Banners Printout

StatPac For Windows						
<u>College Grade Point Average</u>						
N=26	Total	Race			Sex	
		Black	White	Other	Male	Female
		A	B	C	A	B
Total	26 100.0%	7 35.0%	10 50.0%	3 15.0%	11 50.0%	11 50.0%
0.0 - 0.4	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%
0.5 - 1.4	1 4.3%	0 0.0%	1 10.0%	0 0.0%	1 11.1%	0 0.0%
1.5 - 2.4	8 34.8%	4 57.1%	3 30.0%	0 0.0%	5 55.6%	2 20.0%
2.5 - 3.4	11 47.8%	2 28.6%	5 50.0%	2 100.0%	2 22.2%	7 70.0%
3.5 - 4.00	3 13.0%	1 14.3%	1 10.0%	0 0.0%	1 11.1%	1 10.0%

## Alternate Format

When TY=P (the stub is parametric data), each cell of the banners table may contain up to five numbers. To print or suppress any of these numbers, use the AF option.

The parameters for this option are:

- N Number
- M Mean
- S Standard Deviation
- E Standard Error
- D Median

One or more parameters may be used with the AF option. These should not be separated from each other. For example, to print the number, mean, standard deviation and median, use the following option:

```
BANNERS Income By Total Job_Position
OPTIONS TY=P AF=NMSD
..
```

## Category Creation

In most cases you will want the rows and columns of a banners table to be a reflection of the value labels. When CC=L the categories (what StatPac defines as a row or column) will be created from the value labels in the codebook. That is, a row on the stub or column in the banners will be created for each value label in the codebook. If a variable does not have value labels, it will not be included in the table.

In some cases, however, you might not have previously assigned value labels to a variable but still want the variable to be included in the table. Set CC=D to create the categories from the data itself instead of the value labels. For example, if you had a variable with data values of 1-5 but no value labels, you could include this variable

in the banner table by setting CC=D. Alternatively, you could use the LABELS command in the procedure to specify value labels for the variable.

### ***Means & Standard Deviations***

The reserved word MEAN may be used to print the mean average of any row or column variable. The mean will be the average of the value codes (not the value labels). The word MEAN should be included in the variable list immediately following any variable that you want to calculate the mean average of. Either row or column means may be specified. For example, the following command has two variables on each axis (V1 and V2 BY V3 and V4). Means will be printed for both the first and second row variables, and following only the first column variable.

**BANNERS V1 MEAN V2 MEAN BY V3 MEAN V4**

The standard deviations or standard errors may also be printed below the means by specifying the SD=D or SD=E option, respectively. When SD=E and the FP option is used to specify a finite population size, the standard error will be calculated using the finite population correction factor. (See the OPTIONS Command in the Keywords section of this manual.)

### ***Row & Column Totals***

The reserved word TOTAL can be used with the BANNERS command to specify row and/or column totals anywhere in the table. The word TOTAL can be used as a variable name in any position (and as many times as desired) in either (or both) variable lists. When TOTAL is used in the first variable list (for the Y axis), a row is included that displays the totals for all columns in the table. When TOTAL is used in the second variable list (for the X axis), the table includes a column that contains row totals. As an example, a command to print totals in the first row and first column of the banners table would be:

**BANNERS TOTAL, SAT-SCORE, GPA BY TOTAL, CLASS**

A row or column total reflects the number of cases throughout the entire data file in which the value for the row or column appears. Therefore, the numbers for one particular pair of intersecting variables may not add up to the number for the row or column total. For example, if a variable which recorded sex (male/female) is placed on the X axis against a variable on the Y axis which recorded make of car owned, and 20 of the 100 women who completed the survey did not answer the car question, the column total for females would be 100, but the sum of the females in all the rows of the car variable would be only 80.

	Male	Female	Total
Ford	50	20	100
Other	40	60	100
Total	100	100	

It is easy to create a total column that reflects the row totals irrespective of the other cell counts. First use the NEW command to create a new variable called NET. The value of NET will be initialized as missing for all cases. Then use the LABELS

command to assign a label to missing data. Since the banners program uses value labels to determine what is a row and what is a column, it is necessary to use the LABELS command, even though the label is set to blank. Finally, specify the new NET variable instead of the TOTAL keyword in the BANNERS command.

```
NEW (N1) "NET" Totals
LABELS NET (=)
BANNERS TOTAL SCORE BY NET CLASS
```

### **Sort Stub**

The actual labeling for the x and y axes are taken from the study design information in the codebook. In most circumstances, the value labels will reflect the order in which you want the category codes to be listed. It is possible to override the order of the value labels in the study design by using the option (SS=A or SS=D). The category codes (value labels) on the stub will then be displayed in ascending or descending numeric order by frequency.

Additionally, a digit may be added as a suffix to the SS=A or SS=D. It is used to sort the stub excluding the last one or more value labels. This is useful when the last value label is an "other" or "don't know" category, and you want to sort the stub, but still leave the "other" or "don't know" as the last row on the stub. For example SS=D1 would sort the stub in descending order by frequency, except it would leave the last value label as the last row regardless of its frequency.

### **Sort Banners**

The SB option (sort banner) can be used to sort the banner variables. The banner variables to be sorted can be specified, as well as the sort order.

The SB option is enclosed in parentheses and will contain two entries for each banner variable to be sorted. Parameters may be separated by commas or spaces.

The first parameter is A for ascending and D for descending. A number may be used as a suffix to indicate the number of values that should not be included in the sort. For example, if the last value label was "Other" and you wanted it to remain as the last column (even after the other columns were sorted), the first parameter would be D1 for descending order and A1 for ascending order..

You can also indicate whether you want to sort to be done based on the frequencies (counts) or the value label category codes by including a F or C. The default is a sort by frequencies. For example, D2 would indicate a sort by frequencies (the default) in descending order leaving the last two columns not sorted. It could be specified as FD2. Specifying FC2 would be the same sort by category codes.

The second parameter is the variable number (with a V) or the variable name. For example, the following would sort the first (V1) and third (V7) banner variables. The second and fourth banner variables would remain in the order specified by the value labels because they are not specified in the SB option. The first banner variable (V1) would be in descending order by frequency and the third banner variable (V7) would be in ascending order by frequency.

```
Banners StubVar By Total V1 V2 V7 V3
Options SB=(D V1 A V7)
..
```



If you add an exclamation point immediately following the closing parenthesis, the same sorts will apply to all subsequent procedures in this StatPac session.

**Options SB=(D V1 A V7)!**

You can change the sort in a subsequent procedure. You can or cancel it by either closing StatPac, or adding SB=N or SB=() to a procedure

### ***Compress Output***

Compression refers to the way the program creates page breaks. When compression is on (CO=Y), the program will attempt to fit as many columns and rows on a page as possible. That is, page breaks may occur between different value labels of the same variable. When compression is off (CO=N), the program will break pages between each variable on the y-axis. Of course, when there are many categories for a variable, it may be necessary to split up a variable over successive pages regardless of the compression setting. Setting CO=Y or CO=N will apply to both the stub and banner. Compression may be selectively applied to either the stub or banner using CO=S and CO=B, respectively. When compression is set to the stub (CO=S), page breaks will occur between variables (not between value labels), however, the program will still attempt to maximize the number of variables that can fit on a page.

### ***Percentage Base***

The percentage base on a banners analysis can either be the number of respondents (N) or the total number of responses. If PB=N, the denominator for calculating percentages will be the number of respondents. If PB=R, the denominator will be the total number of responses for all individuals.

### ***Special Value Label HIDE***

When creating a banner table, it is often desirable to display only some of the response categories. The LABELS command may be used to eliminate undesirable categories from the table. This will eliminate the column from the table and from any calculations of percentages on the table.

For example, assume the following counts for V1:

1	2	3		
Agree	Neutral	Disagree	No Response	Total N
30	20	40	10	100

If PB=N, (denominator equals number of respondents), the percents will be:

Agree	Neutral	Disagree
30%	20%	40%

If PB=R, (denominator equals number of responses), the percents will be:

Agree	Neutral	Disagree
30/90=33%	20/90=22%	40/90=44%

We could use the following LABELS command to eliminate the "Neutral" category from the table:

```
LABELS V1 (1=Agree)(3=Disagree)
```

If PB=N, the percents will still be based on a denominator of 100. If however, PB=R, then the percents will be based on a denominator of 70 (30+40):

Agree	Disagree
30/70=43%	40/70=57%

The special value label "HIDE" may be used to suppress printing of a value label without reducing the denominator for the percents calculations. The following LABELS command could be used to eliminate the "Neutral" category from the table, while still including the "Neutral" count in the denominator:

```
LABELS V1 (1=Agree)(2=Hide)(3=Disagree)
```

Any row or column that has a value label of "HIDE" will not be printed, but it will be included in the percent calculations when PB=R. Note that the percentages are based on the counts for all value labels (including the "Neutral" category), even though all the value labels are not displayed in the table.

Agree	Disagree
30/90=33%	40/90=44%

If you only wanted the "Agree's" to show in the table, you could use the following statements in the procedure. The percentages in the table would still be based on 90:

```
LABELS V1 (1=Agree)(2=Hide)(3=Hide)  
OPTIONS PB=R
```

..

## ***Multiple Response***

Multiple response variables can be included in banner crosstabs by using the MR option to combine those variables that should be interpreted as a single variable. The syntax to combine multiple response variables is:

```
OPTIONS MR=(<list 1>)(<list 2>)...(<list n>)
```

Each variable list represents a group of multiple response items that should be grouped as if they were a single variable. Each group must be enclosed in parentheses and specified as a variable list (individual variables are separated by

commas or spaces, and ranges are specified with a dash). Either variable names or variable numbers may be specified in the MR option variable list. The V prefix is optional for variable numbers.

The sequence of variables specified in a multiple response group list must match a sequence of the x or y axis banner list. For example, consider the following BANNERS command:

**BANNERS V5 - V8, V10, V11 BY V1 - V3, V5, V6**

Multiple response variables    y-axis: 5-8,10,11    x-axis: 1-3,5,6

OPTIONS MR=(1-3)	Groups vars. 1, 2 & 3 on the x-axis
OPTIONS MR=(2,3)	Groups vars. 2 & 3 on the x-axis
OPTIONS MR=(8,10-11)	Groups vars. 8, 10 & 11 on the y-axis
OPTIONS MR=(5,6)	Groups vars. 5 & 6 on the x & y-axis

The following groupings might cause problems:

OPTIONS MR=(1-6)	Groups vars. 1, 2 & 3 on the x-axis, but not variables 5 or 6 because variable 4 was not part of the banners variable list
OPTIONS MR=(6-11)	Groups vars. 6, 7 & 8 on the y-axis, but not variables 10 or 11 because variable 9 was not part of the banners variable list
OPTIONS MR=(10-15)	Groups vars. 10 & 11 on the y-axis and ignores the extra variables
OPTIONS MR=(3-7)	No variables grouped
OPTIONS MR=(11-20)	No variables grouped
OPTIONS MR=(3,2,1)	No variables grouped
OPTIONS MR=(8,11,10)	No variables grouped

In general, the MR option will never cause a fatal error. If an invalid grouping is found, it is simply ignored and the variables will not be grouped on the output. The banners program uses the value labels from the first variable specified in each group list. The MR option should be used only to group variables which share a list of common value labels. The value labels must be specified in the study design (i.e., they will not automatically be determined from the data file). This was implemented to prevent misspelled or spurious data from creating its own row or column in the output.

### **Net Codes**

Net categories may be created and displayed on the stub using the NT option in conjunction with the MR option. The NT option specifies the codes on the stub variable that are to be interpreted as net categories. Net categories are excluded from

the calculations of totals and means. Multiple categories are separated with a slash and enclosed in quotes. The general format is:

```
OPTIONS NT="code/code/code"
```

For example, suppose you want to create a banner table where the stub (V5) is a five-point Likert scale. The scale is coded: (1=Very good) (2=Good) (3=Fair) (4=Poor) (5=Very poor). You want the stub to contain two net variables and look like this:

```
1=Very Good
2=Good
NET: Very Good or Good
3=Fair
NET: Poor or Very Poor
4=Poor
5=Very Poor
Mean & SD
```

The first step is to create a NET variable and compute it equal to values not used in the originally coded variable. In this example, 6, 7, 8, 9, and 0 are unused, so we could use any of them for the new NET variable. Then use the LABELS command to relabel the stub categories in the order you want them to appear. Include the new NET variable in the BANNERS command as if it were a multiple response variable. Use the MR option to specify multiple response and use the NT option to specify which codes are the net categories.

```
New (N1) "NET"
If V5="1/2" Then Compute NET=6
If V5="4/5" Then Compute NET=7
Labels V5 (1=Very Good) (2=Good) (6=NET: Very Good or Good)
(3=Fair) (7=NET: Poor or Very Poor) (4=Poor) (5=Very Poor)
Banners V5 NET Mean By Total Age Gender Group
Options MR=(V5 NET) NT="6/7"
..
```

## Weighting

Weighting is useful when the true incidence in the population is known, but data collection yielded a different incidence. In other words, there was a sampling error (the sample does not adequately represent the population). Weighting can be used to mathematically increase or decrease the counts of any banner variables so they more accurately reflect the known population parameters.

The WEIGHT command in StatPac will create a weighted file using integer case weights where a probability function is used for the non-integer portion of the weights. The WT option in the BANNERS command will not create a new data file, but rather, simply adjusts the counts in the banner table. The WEIGHT command and the WT option in banners are different methods of accomplishing the same goal and should not be used together in the same procedure.

The easiest way to create a weighting variable is to use the Utility program, Sampling > Create Variable for Weighting. It will allow you to create a weighting variable that can be used in any of the StatPac procedures. If you're weighting by more than one variable simultaneously, the utility program is the best way to create the weights.

The following explanation is if you want to manually create a weighting variable. It is presented here to show the mathematical concepts involved in weighting. While you can create a weighting variable manually, we recommend using the utility program.

Take for example a simple banner table with an automatic total row and a mean row:

Title (#)  
 Banners V1 By Total Gender  
 Options AT=Y AM=Y PC=Y  
 ..

The table might look like this:

N=48	Total	Gender	
		Male	Female
Total	48 100.0%	31 64.6%	17 35.4%
1=Very Good	10 20.8%	9 29.0%	1 5.9%
2=Good	12 25.0%	8 25.8%	4 23.5%
3=Fair	18 37.5%	12 38.7%	6 35.3%
4=Poor	6 12.5%	1 3.2%	5 29.4%
5=Very Poor	2 4.2%	1 3.2%	1 5.9%
Mean	2.54	2.26	3.06
SD	1.09	1.03	1.03

Looking at the Total row, we see that our sample had 64.6% males and 35.4% females. However, we know that the population actually has 55% males and 45% females, so the Total column might be producing an inaccurate reflection of the total population due to a sampling error. To correct the problem, we would weight the gender variable so the table reflects the 55% and 45% proportions that we know exist in the population.

The first step is to calculate the weights for males and females. The weights are easily calculated by the following formula:

$$\text{Weight} = \text{Desired Percentage} / \text{Observed Percentage}$$

Thus, the weight for males would be 55 divided by 64.6 = .8514 and the weight for females would be 45 divided by 35.4 = 1.2712.

Typically, you'll want to weight the entire banner table. Begin by creating a variable that contains the weight for each case. Subsequent procedures would specify the WT option to weight the entire banner tables by the case weight variable

```
STUDY SEGMENT
NEW (N7) "CaseWeight"
IF GENDER = 1 THEN COMPUTE CaseWeight = 0.8514
IF GENDER = 2 THEN COMPUTE CaseWeight = 1.2712
SAVE
..
Banners V1 By Total Age Gender Group
Options WT=(CaseWeight)
..
Banners V11-V20 By Total Age Gender Group
Options WT=(CaseWeight)
..
```

The other form of the WT option lets you weight individual banner variables with their own weights

The first step is to calculate the weights for males and females using the same formula as above. The next step is to add the WT option to the banners procedure. The format for the WT option is:

```
OPTIONS WT=(variable code=weight code=weight)
```

Spaces or commas may be used within the parentheses to separate each of the components of the option.

In this example, the codebook specifies 1=Male and 2=Female so the WT options would use codes of 1 and 2.

```
Title (#)
Banners V1 By Total Gender
Options AT=Y AM=Y PC=Y WT=(Gender 1=.8514 2=1.2712)
..
```

Rerunning the procedure would produce a weighted analysis with an adjusted total row and total column.

N=48	Total	Gender	
		Male	Female
Total	48 100.0%	26 55.0%	22 45.0%
1=Very Good	9 18.6%	8 29.0%	1 5.9%
2=Good	12 24.8%	7 25.8%	5 23.5%
3=Fair	18 37.2%	10 38.7%	8 35.3%
4=Poor	7 15.0%	1 3.2%	6 29.4%
5=Very Poor	2 4.4%	1 3.2%	1 5.9%
Mean	2.62	2.26	3.06
SD	1.09	1.03	1.02

More than one banner variable may be weighted. The syntax is the same except additional sets of parentheses are added for each variable to be weighted.

OPTIONS WT=(variable code=weight code=weight) (variable code=weight code=weight code=weight)

When the WT option is used, the total column will reflect the weighted values of the variable that follows it. If more than one variable is weighted, it would be wise to specify more than one total column. For example, if ethnicity were coded as an alpha variable (W=White and B=Black), the following commands would produce a total column for gender and a total column for ethnicity, and both would be weighted:

Banners V1 By Total Gender Total Ethnicity

Options AT=Y AM=Y PC=Y WT=(Gender 1=.8514 2=1.2712)(Ethnicity W=.5672 B=1.8141)

..

### ***Fractional Counts***

The FC option may be used in conjunction with the WT option to display fractional cell counts. FC=Y will show the decimal portion of the cell counts and FC=N will display them as integers. While weighting does create fractional cell counts, it is often confusing (e.g., how could there be 178.6 males?). Using FC=N will round all cell counts to whole numbers, while FC=Y will show the decimal portions.

## ***Supplemental Heading***

The supplemental heading is a line of text that will appear after the heading and title, but before the banner table. It may contain any text and should be enclosed in quotes. When the pound symbol is used in the supplemental heading, it will be printed as the number of cases. The SH option is usually used to indicate who is included in the banner table. The following is an example if a supplemental heading:

OPTIONS SH="BASE: ALL RESPONDENTS (N=#)"

## ***N Equals***

The sample size can be displayed in the top left corner of the table with the NE option. It may contain any text and should be enclosed in quotes. When the pound symbol is used in the N Equals option, it will be printed as the number of cases. The NE option is usually used to indicate who is included in the banner table. The following is an example of the N Equals option:

OPTIONS NE="(N=#)"

## ***Significance Tests***

StatPac offers significance testing in banner tables. To bypass all significance testing, set the ST option to none (ST=N). The following options control the type of significance tests:

OPTIONS ST=N (no significance tests)

OPTIONS ST=P (t-test between percents only)

OPTIONS ST=M (t-tests between means only)

OPTIONS ST=T (t-tests between percents and means)

OPTIONS ST=C (chi-square tests for each subtable)

OPTIONS ST=A (t-tests between means and percents and chi-square tests)

## ***T-Tests Between Proportions and Means***

Two-tailed t-tests between column percents and means can be performed with the ST option. When specified, StatPac will automatically set the banner to include a code letter for each column, beginning with column "A". An independent samples t-test will be performed between all combinations of banner columns, and the results will be displayed in the table if they are significant at the alpha levels set by the C1 and C2 options.

Upper case letters indicate "high significance: and lower case letters indicate "moderate significance" (high and moderate being defined by the values of C1 and C2). For example, suppose C1=.05 and C2=.01. After running the analysis, you see a cell with the letters "Ce". This means that the percentage in this cell is significantly different from the percentage in column C at the .01 level, and significantly different from the percentage in column E at the .05 level.

## ***Chi-Square Tests***

Banner crosstab tables may be broken down into several combinations of smaller tables, consisting of one variable on each axis. For example, the following BANNERS statement could be broken down into three subtables:



## BANNERS V1 BY V2, V6, V9

The subtables would be V1 by V2, V1 by V6, and V1 by V9. It is then possible to calculate a chi-square statistic for each subtable. Use the option ST=C to request a chi-square analysis for all the combinations of subtables. The chi-square, degrees of freedom and probability of chance will be printed for each subtable.

It is not possible to calculate chi-square statistics for tables with completely missing rows or columns; therefore, if any row or column in a subtable is completely missing, it will not be included in the calculation of the chi-square statistic or degrees of freedom (even though it may be displayed in the count/percent table).

### ***Example of a Two-Way Chi-Square Statistics Printout***

<b>StatPac For Windows</b>			
<b><u>College Grade Point Average</u></b>			
<u>Chi-square statistics for all variable pairs</u>			
<u>Rows -by- Columns</u>	<u>Chi-Square</u>	<u>DF</u>	<u>Probability</u>
College Grade Point Average -by- Race	4.560	6	.601
College Grade Point Average -by- Sex	5.025	3	.170

When ST=A, all types of significance testing will be performed. The output will include the t-tests between percents and means and two-way chi-square tests.

### ***Yate's Correction***

If degrees of freedom equals one (i.e., when the banners program produces a two-by-two table), the chi-square statistic can have the Yate's correction applied. The option YA=Y will enable Yate's correction for two-by-two tables, while YA=N will disable it.

### ***Zero Rows & Columns***

You may choose whether or not to print zero rows and columns. This situation (of zero rows or columns) could occur if there are value labels in the study design for which there is no data. If you want the reader of your report to know that a category exists, you will probably want to print rows and columns with zero counts (ZR=Y ZC=Y). In most cases, however, conserving space is more important, so you would set ZR=N and ZC=N.

### ***Automatic Page Title Creation***

When performing a series of banner analyses, each having the same banner columns, and only one y axis variable (per page), it may be desirable to make the page title the same as the y axis variable label. When the title is set to a pounds symbol in parentheses, the title will become the variable label for the y axis variable. (This can be changed to the x axis label using a patch).

For example, let's say you had several demographic variables as your banner points, and you wanted to look at several other variables on the y axis (down the stub). You want a series of tables that look like this:

Special Study (Page Heading)

The variable label of the "Some Variable" on the y axis (Title)

	Age	Sex		Income		
	<u>Under 21</u>	<u>Male</u>	<u>Female</u>	<u>Low</u>	<u>Middle</u>	<u>High</u>
Some	-----	-----	-----	-----	-----	-----
Variable	-----	-----	-----	-----	-----	-----

The following procedure would produce five similar tables, each on a different page, and each with a different title:

```
STUDY Yourstudy
HEADING Special Study
TITLE (#)
BANNERS V1-V5 BY AGE SEX INCOME
OPTIONS CO=N SH=""
..
```

### **Total Row Position**

When the TOTAL keyword is imbedded between other variables in the banners command line, the TP option is used to determine whether the total should be printed for the previous variable or the next variable. In the following example, the total row could be the last stub for V1 or it could be the first stub for V2, depending on the setting of the TP option.

```
BANNERS V1 TOTAL V2 BY AGE SEX INCOME
```

If TP=L, the last row for V1 will be a total row. If TP=F, the first row for V2 will be a total row.

### **Total Counts**

The TC option makes it possible to print only the counts (without the percents) in total rows and total columns, even when percentages are being printed in the rest of the table. If TC=Y, total rows and total columns will only contain the counts. If TC=N, total rows and columns will be defined by the PF option, and will contain the same number of values as the other cells in the table.

### **Total Adjustment**

The TA option may be used to set how total columns are calculated. If TA=N, the total columns will be based on the number of non missing cases for the stub variable. If TA=Y, the total column counts will be the sum of the counts for the variable that follows it. If there are no missing data for the banner variable, the counts will be the

same, but if the banner variable that follows the TOTAL keyword has missing data, the counts in the total column will be different. Thus, when setting TA=Y, you could insert the word TOTAL before each banner variable and each total column might contain different counts.

### ***Total Row Denominator***

Normally, a total row will be based on the same denominator as specified by the PB option (either N the number of cases, or R the number of respondents). If PB is set to R, you can force the percentages in a total row to be calculated using N as the denominator by setting TD=Y. This is sometimes handy when the banner contains multiple response variables.

### ***Total Total Intersections***

When printing a table that contains both total rows and total columns, there will be at least one intersection of a total row and a total column. You must set the precedence as to how the intersection cell is calculated. It may be based on the sum of the row counts (TT=R) or the sum of the column counts (TT=C).

### ***Automatic Total Row***

The AT option may be used to automatically print a total row for each variable on the stub. Its purpose is to eliminate the necessity of having to type the TOTAL keyword for each of the stub variables. In the previous example, if you wanted each stub variable to begin with a total row, the command would be:

```
BANNERS TOTAL V1 TOTAL V2 TOTAL V3 TOTAL V4 TOTAL V5  
BY AGE SEX INCOME  
OPTIONS CO=N  
..
```

If the AT option is set to "Y", the total rows will be included in the output, even when the TOTAL keyword is not included in the stub variable list. The following procedure would produce the same output as the previous procedure:

```
BANNERS V1-V5 BY AGE SEX INCOME  
OPTIONS CO=N AT=Y TR=F  
..
```

The TR option is used in conjunction with the AT option to determine whether the total row will be the first or last row on the stub. Note that if you use the option AT=Y, then you should not use the TOTAL keyword anywhere in the stub variable list.

### ***Automatic Mean Row***

The AM option may be used to automatically print a row of means for each variable on the stub. Its purpose is to eliminate the necessity of having to type the MEAN keyword for each of the stub variables. In the previous example, if you wanted each stub variable to include a row of means, the command would be:

```
BANNERS V1 MEAN V2 MEAN V3 MEAN V4 MEAN V5 MEAN BY  
AGE SEX INCOME  
OPTIONS CO=N
```

..

If the AM option is set to "Y", a row of means will be included in the output, even when the MEAN keyword is not included in the stub variable list. The following procedure would produce the same output as the previous procedure:

BANNERS V1-V5 BY AGE SEX INCOME  
 OPTIONS CO=N AM=Y

..

## ***Labeling and Spacing Options***

Option	Code	Function
Labeling	LB	Sets the column headings to print the variable label (LB=E), the variable name (LB=N), or the variable number (LB=C).
Labeling Width	LW	Set the maximum width (in inches) for the variable labels on the stub.
Exact Width	EW	When EW=Y, the labeling width for the stub will be exactly what is specified with the LW option. When EW=N, the width of the stub will self-adjust based on the length of the stub labels.
Column Width	CW	Sets the minimum width of the columns (in inches).
Column Spacing	CS	Sets the spacing (in inches) between the columns.
Variable Spacing	VS	Sets the spacing (in inches) between the banner variables.
Key	KY	Sets whether the top left corner of the banner will show a legend of the cell contents.
Print Percent Symbol	PP	Sets whether percentage symbols will be shown.
Decimal Places	DP	Sets the number of decimal digits that will be shown.
Print Codes	PC	Sets whether the code (to the left of the equals symbol) will be shown with the value labels.
Underline Stub Variable Labels	UL	Sets whether the stub variable labels will be underlined.
Bottom Justify	BJ	Sets whether the banner labels will be bottom justified.
Heading Justification	HJ	Sets the justification for the banner variable labels
Bottom Justify Heading	BH	Sets whether the banner variable labels will be bottom justified.
Value Label Justification	LJ	Sets the justification for the banner value labels.
Bottom Justify Value Labels	BL	Sets whether the banner labels will be bottom justified.
Extra Spacing	ES	When ES=Y, a blank line will be printed above and below the banner value labels. When ES=N, no blank lines will be printed.
Code Justification	CJ	Sets the justification for the value label codes.
Print Stub Variable Label	SL	Sets whether the stub variable labels are shown.
Stub Variable Spacing	VY	Sets the number of blank rows between variables

		on the stub.
Stub Label Spacing	LY	Sets the number of blank rows between value labels on the stub.

### ***Minimum Cell Count***

Researchers often choose not to show the percentages for cells containing a small N. The MC option may be used to suppress the printing of percentages of cells with low counts. For example, if MC=5, StatPac will print cell counts that are greater than or equal to 5. If a cell has a count of less than 5, StatPac would print dashes instead of the percent. If MC=1, StatPac will print dashes for cells where the count is zero. Valid values for MC are between 0 and 100. If MC=0, all cell counts will be printed.

### ***Minimum Denominator***

Percentages can be misleading if they are based on a small denominator. The MD option may be used to suppress the printing of percentages that are based on a small denominator. The MD option sets the minimum denominator that StatPac will use for calculating percents. For example, if MD=5, StatPac will calculate percentages if the denominator is greater than or equal to 5. If a denominator were less than 5, StatPac would print dashes instead of the percent. Valid values for MD are between 0 and 100. If MD=0, all percentages will be printed.

---

## **DESCRIPTIVE Command**

Descriptive statistics are usually the first step in the analysis of interval or ratio data. They reveal central tendency and the shape of the distribution.

The syntax of the command to run descriptive statistics is:

**DESCRIPTIVE** <Variable list>

For example, if you are examining college entrance exam scores for READING (V7), ARITHMETIC (V12) and VERBAL (V19) skills, descriptive statistics could be requested with any of the following commands:

**DESCRIPTIVE READING, ARITHMETIC, VERBAL**  
**DESCRIPTIVE V7, V12, V19**  
**DE V7 V12 V19** (DESCRIPTIVE may be abbreviated as DE)

There are a wide variety of descriptive statistics available. To print or exclude individual statistics, use the appropriate option.

Missing data (blanks) will always be excluded from the calculation of descriptive statistics. It will be reported as the number of missing cases but will not be used for any calculations.

## One Analysis

The one-analysis option allows you to print selected descriptive statistics for several variables on one page. This option is especially useful for summary reporting, when you only need a few descriptive statistics for a large number of variables.

All the variables specified with the OA option must be numeric. For example, suppose that variables 25-34 are ten numeric scores. The following commands would produce a one-page summary of selected descriptive statistics for each of the ten items:

```
DESCRIPTIVE V25-V34
OPTIONS OA=Y
```

### Example of a Descriptive Statistics One-Analysis Printout

StatPac For Windows					
<u>Freshman Class - 3rd Quarter - 1990</u>					
	Mean	SD	Total	Median	99% CI
Reading	60.80	17.63	25	61	51.72 - 69.88
Writing	60.00	19.02	25	60	50.21 - 69.79
Verbal	60.00	14.31	25	62	52.63 - 67.37
History	55.20	19.56	25	51	45.13 - 65.27
Math	54.40	19.29	25	50	44.46 - 64.34
Science	50.80	17.96	25	52	41.55 - 60.05

When the OA option is used, the keyword RECORD in the variable list will leave a blank line in the report. This allows you to visually group variables. For example, this will print five rows of descriptive statistics, a blank row, and then five more rows.

```
DE v1-v5 Record V6-v10
OPTIONS OA=Y
```

..

## Statistics

When the OA option is specified, you may select which descriptive statistics you want with the ST option. The codes for the ST option are the same as the specific statistic codes described below. (The only exception is NC, which stands for the number of valid cases). For example, the following commands would report the mean, median, unbiased standard deviation and number of cases for variables 25-34:

DESCRIPTIVE V25-V34  
OPTIONS OA=Y ST=(ME MD US NC)

Note that the commands are identical to the previous example except the ST option is used to identify the specific statistics you want calculated. The parentheses around the list of statistics is mandatory.

### **Sort Variables**

When performing descriptive statistics with the one-analysis option (OA=Y), you can sort the variables by the contents of the first column of the results. The SV (sort variables) option may be set to "N" for no sort, "A" to sort in ascending order, or "D" to sort in descending order. When no sort is specified, the variables will be listed in the order that they appear in the analysis command variable list. The SV option is applicable only when the OA=Y option is specified.

Additionally, a digit may be added as a suffix to the SV=A or SV=D. It is used to sort the variables excluding the last one or more variables. This is useful when the last variable is an "other" variable, and you want to sort the variables, but still leave the "other" as the last variable. For example SV=D1 would sort the variables in descending order, except it would leave the last variable as the last row regardless of its value.

### **Minimum, Maximum, Range, & Sum**

There are four very simple measures of dispersion that give an overall picture of the data. These are the minimum data value, maximum data value, range (maximum minus the minimum), and sum of the data. An option line that would enable all of these features is:

OPTIONS MI=Y MA=Y RA=Y SU=Y

### **Mean, Median, & Mode**

The best known descriptive statistics are the mean, median and mode. They describe the central tendency of a distribution. The mean (average) is the most popular. It is found by adding the values for all the (non-missing) cases and dividing by the number of (non-missing) cases. For example, to find the mean age of all your friends, add all their ages together and divide by the number of friends. The mean average can present a distorted picture of central tendency if the sample is skewed in any way.

For example, let's say five people take a test. Their scores are 10, 12, 14, 18, and 94. (The last person is a genius.) The mean would be the sums of the scores 10+12+14+18+94 divided by 5. In this example, a mean of 29.6 is not a "good" measure of how well people did on the test in general. When analyzing data, be careful of using only the mean average when the sample has a few very high or very low scores. These scores tend to skew the shape of the distribution and will distort the mean.

The median provides a measure of central tendency such that half the sample will be above it and half the sample will be below it. For skewed distributions this is a better measure of central tendency. In the previous example, 14 would be the median for the sample of five people.

The mode is the most common score or category - the one which occurred most frequently. It is possible to have more than one mode if there is not a single "most frequent score". For example, the following set of data has two modes: 12 and 16.

12 12 12 13 14 15 15 16 16 16 17 18

The distribution of many variables follows that of a bell-shaped curve. This is called a "normal distribution". One must assume that data is approximately normally distributed for many statistical analyses to be valid. When a distribution is normal, the mean, median and mode in the population will all be equal. If they are not equal, the distribution is distorted in some way.

### ***Skewness, Kurtosis, & Kolmogorov-Smirnov***

There are basically two ways that a distribution can be distorted: skewness and kurtosis. Skewness refers to "top heavy" or "bottom heavy"; (i.e., the tail of the curve). If the longest tail of the curve goes to the right (the curve is top heavy), it is positively skewed. If it is bottom heavy (the longest tail of the curve goes to the left), it is negatively skewed. A value of zero for skewness represents a symmetrical distribution, such as the normal distribution mentioned above.

Kurtosis refers to how peaked or flat the curve is. A very flat curve is called "platykurtic" and has a kurtosis of less than three. A very peaked curve is called "leptokurtic" and has a kurtosis greater than three. A value of three for kurtosis indicates normal peakedness and the distribution is termed "mesokurtic".

The Kolmogorov-Smirnov statistic provides a quick check to determine the degree of normality in the data. The value provides a relative indication of normality; as the value moves further away from zero, we can be more certain that the data does not approximate a normal distribution. The distribution is non-normal:

at the .15level if  $KS > .775$   
at the .10level if  $KS > .819$   
at the .05level if  $KS > .895$   
at the .025 level if  $KS > .955$   
at the .01level if  $KS > 1.035$

### ***Standard Deviation & Variance***

The standard deviation is a very useful statistic that measures the dispersion of scores around the mean. On the average, 68 percent of all the scores in a sample will be within plus or minus one standard deviation of the mean and 95 percent of all scores will be within two standard deviations of the mean.

The variance is calculated directly from the distribution of raw scores. It is the sum of the squared deviations of each score from the arithmetic mean divided by N. The standard deviation is simply the square root of the variance. The unbiased estimates should be used when sampling from the population. The formula for the unbiased estimates of the variance and standard deviation is the same except that N-1 is used in the denominator.



## ***Standard Error & Confidence Intervals***

Confidence intervals are very important. They allow us to predict where the mean would fall if another sample is taken. The standard error of the mean is used to estimate the range within which we would expect the mean to fall.

Let's say the 95 percent confidence interval for the mean is 12.4 to 22.8. In repeated samples of the same size, the mean would be expected to fall between these two values 95 percent of the time. A similar interpretation can be made for the 99 percent confidence interval. The 95 and 99 percent confidence intervals may be requested using the C5 and C9 options respectively:

**OPTIONS C5=Y C9=Y**

The above formula for the standard error of the mean is used when the sample size is small relative to the population size (say, less than ten percent). When the sample size represents a substantial proportion (greater than ten percent) of the population, the standard error is modified by the finite population correction factor. This has the effect of reducing the standard error and narrowing the confidence interval band. When the FP option is used to specify a population size, the standard error will be adjusted and printed as the "Corrected Standard Error Of The Mean". (See the OPTIONS command in the Keywords section of this manual for information on using the FP option.)

Confidence intervals are accurate only if the distribution of the data resembles a normal curve. Be careful; using confidence intervals from non-normal data is risky business.

## ***Example of a Descriptive Statistics Printout***

## StatPac For Windows

### Score On Freshman Biology Test

Minimum = 0

Maximum = 57

Mean = 19.39

Median = 5

Mode = 5

Variance (Unbiased) = 572.07

Standard Deviation (Unbiased) = 23.92

Corrected Standard Error Of The Mean = 4.55

95 Percent Confidence Interval Around The Mean = 10.48 - 28.30

Valid Cases = 23

Missing Cases = 0

Response Percent = 100.0%

Population Size = 175

### ***Quartiles & General "-iles"***

Quartiles are often used in education to divide a distribution into 4 groups of equal N. A quartile printout will contain three values (one less than the number of groups). If, for example, the value for the first (lowest) quartile is 50, it means that 25% of the sample had a score of 50 or less. You can specify any division with the IL option.

For example, if you specify IL=10, then deciles will be printed. If the ninth decile (highest) value is 85, it means that 90% of the distribution had a score of 85 or less, and 10% scored equal to or higher than 85. The "-ile" values are interpolated when necessary. Set IL=1 to disable the "any iles" option

### ***Example of a Quartile Printout***

#### Quartiles

1 = 2

2 = 4.50

3 = 51.25

### ***Labeling and Spacing Options***

Option	Code	Function
Labeling	LB	Sets the labeling to print the variable label (LB=E), the variable name (LB=N), or the variable number (LB=C).
Labeling Width	LW	Set the maximum width (in inches) for the variable

		labels on the stub.
Exact Width	EW	When EW=Y, the labeling width for the stub will be exactly what is specified with the LW option. When EW=N, the width of the stub will self-adjust based on the length of the stub labels.
Column Spacing	CS	Sets the spacing (in inches) between the columns.
Decimal Places	DP	Sets the number of decimal digits that will be shown.
Label Justification	LJ	Sets the justification for the banner variable label.
Extra Spacing	ES	When ES=Y, a blank line will be printed between each variable on the stub. When ES=N, no blank lines will be printed.

---

## BREAKDOWN Command

The breakdown program gives descriptive statistics for one or more criterion variables broken down by one or more subgroup variables. In other words, the breakdown program provides a way of summarizing descriptive statistics for many subgroups. The same information could be obtained by performing multiple descriptive statistics analyses using the IF-THEN-SELECT command to limit each analysis to the desired subgroup.

The syntax for the command to evoke the breakdown program is:

**BREAKDOWN <Criterion var. list> BY <Subgroup var. list>**

For example, let's say you want descriptive statistics for AGE; however, you want these statistics broken down by RACE, SEX and INCOME level. In other words, you are interested in comparing age for each of the subgroups (e.g., average age of males versus average age of females).

A data file for this analysis would look like this:

```
AM136  (record 1 - race is coded as A
        sex is coded as M
        income level is coded as 1
        age is 36)
BF342  (record 2 - race is coded as B
        sex is coded as F
        income level is coded as 3
        age is 42)
```

The criterion variable is AGE (V4). It is this variable that you will be calculating descriptive statistics for, so it must be interval or ratio-type data.

Up to ten subgroup variables may be included in the subgroup variable list. These variables may be either alpha or numeric. In our example, these would be: RACE (V1), SEX (V2) and INCOME (V3). Each of the subgroup variables may contain up to 100 categories (value labels).

Any of the following commands would perform the analysis:

BREAKDOWN AGE BY RACE, SEX, INCOME  
BREAKDOWN AGE BY RACE - INCOME  
BREAKDOWN AGE BY RACE SEX INCOME  
BR V4 BY V1 - V3 (BREAKDOWN may be abbreviated as BR)

Notice that the keyword BY is mandatory. This is necessary because you may want a breakdown on several criterion variables. That is, several different variables may be broken down by the same subgroup variables.

When a criterion variable list is specified, it is equivalent to performing a different breakdown for each criterion variable. For example, both AGE and IQSCORE could be broken down by RACE, SEX and INCOME:

BREAKDOWN AGE IQSCORE BY RACE SEX INCOME

When a criterion variable list (AGE and IQSCORE) is specified, it is the same as requesting a separate analysis for each variable in the list. In this example, two tasks will be performed. They are:

BREAKDOWN AGE BY RACE SEX INCOME  
BREAKDOWN IQSCORE BY RACE SEX INCOME

When specifying a criterion variable list, care must be taken to insure that each variable in the criterion variable list is different from those in the subgroup variable list. That is, a variable cannot be broken down by itself.

The output from the breakdown program will print the mean, standard deviation, number of cases, and percent for each of the subgroups.

### Example of a Breakdown Printout

StatPac For Windows				
<u>Test Score</u>				
	Mean	SD	N	Pct.
For Entire Sample (Missing = 2)	45.43	13.12	23	92.0%
<u>Sex</u>				
Male	42.91	11.02	11	47.8%
Female	49.00	16.14	10	43.5%
Missing	41.50	0.71	2	8.7%
<u>Ethnic Origin</u>				
White	38.56	11.18	9	39.1%
Black	51.13	14.06	8	34.8%
Other	49.60	12.38	5	21.7%
Missing	41.00	0.00	1	4.3%

### Sort Type & Sort Order

The output from the breakdown analyses may be more meaningful when the subgroup categories are displayed in sorted order. If no sort is selected (ST=N), the subgroup categories will be displayed in the order they appear in the study design. If the study design does not contain all the values in the data file (such as misspunched data), the unlabeled values will appear on the printout in the order that they are encountered in the data file.

You can sort the subgroup categories by frequency of response using the option ST=F, or by the category codes themselves (ST=C). For example, the following option would sort the categories by frequency of response in descending order. It would be requested with the following options:

OPTIONS ST=F SO=D (Sort Type by frequency of response)  
(Sort Order is descending)

In most cases, you'll probably want to have the breakdown printout appear in ascending order by the code. The options statement to do this is:

OPTIONS ST=C SO=A (Sort Type is by category code)  
(Sort Order is ascending)

Notice that this type of sort is generally the way the information would be listed in the study design. If this is the case, sorting by category code will have no effect. Sorting by category codes is useful if you did not enter value labels for the subgroup variable.

### Print Missing

When a subgroup variable is missing, it may be included or excluded from the analysis with the PM option. When PM=Y, all subgroup variables that are missing

will be grouped into a unique category and descriptive statistics for the criterion variable will be reported for the "missing category".

### **Category Creation**

Sometimes there may be a subgroup category listed in the study design that has no accompanying data. For instance, nobody in the sample may be over 60 years old. Whether or not you want the label to appear with a count of zero is a matter of preference.

The actual categories (value labels) in the breakdown analysis can be created either from the study design value labels (CC=L) or from the data itself (CC=D). When the categories are created from the labels, the value labels themselves will be used to create the categories for the analysis, and data that does not match up with a value label code will be counted as missing. When categories are created from the data, all data will be considered valid whether or not there is a value label for it.

### **Percentage Base**

In addition to means and standard deviations, the breakdown analysis also prints counts and percents for each of the categories. The denominator for the percentages can either be the number of respondents or the total number of responses. If PB=N, the denominator for calculating percentages will be the number of respondents (i.e., the number of records in the data file). If PB=R, the denominator will be the total number of responses for all individuals.

### **Labeling & Spacing Options**

Option	Code	Function
Labeling	LB	Sets the column headings to print the variable label (LB=E), the variable name (LB=N), or the variable number (LB=C).
Labeling Width	LW	Set the maximum width (in inches) for the variable labels on the stub when OA=Y.
Exact Width	EW	When EW=Y, the labeling width for the stub will be exactly what is specified with the LW option. When EW=N, the width of the stub will self-adjust based on the length of the stub labels.
Column Spacing	CS	Sets the spacing (in inches) between the columns when OA=Y.
Decimal Places	DP	Sets the number of decimal digits that will be shown.
Print Percent Symbol	PP	Sets whether the percent symbol is shown.
Print Codes	PC	Sets whether the codes are printed with the value labels.
Label Justification	LJ	Sets the justification for the banner variable label when OA=Y.
Extra Spacing	ES	When ES=Y, a blank line will be printed after each variable name or label on the stub. When ES=N, no blank lines will be printed.

---

# TTEST Command

The t-test is a relatively simple statistic to test the difference between two means even when the sample sizes are small (less than 30). The two variables must be interval or ratio-type data. StatPac lets you test the difference if the N's are equal or unequal. The primary advantage of the t statistic is that it allows us to test the difference between samples with small numbers of cases.

The t distribution depends on the size of the samples. With small samples, the t distribution is leptokurtic; however, as the sample size exceeds 30, the t distribution approaches that of the normal curve. The standard error of the difference is used to establish a range where the difference between the true means of the two populations would be expected to fall.

The significance of the t statistic depends upon the hypothesis the researcher plans to test. This hypothesis should be developed before collecting the data. If interested in determining whether there is a significant difference between two means, but you do not know which of the means is greater, use the two-tailed test. If interested in testing the specific hypothesis that one mean is greater than the other, use the one-tailed test.

There are two basic kinds of t-tests; one for matched pairs and the other for independent groups.

## T-Test For Matched Pairs

If each subject or unit being tested was measured in both groups, then the appropriate t-test is for matched pairs. To perform this type of analysis, you must enter the data so that both observations for a subject are in the same data record. An example of an appropriate use of the t-test for matched pairs might be to compare pretest and posttest scores where each person took both a PRETEST (V1) and a POSTTEST (V2). Both values are contained in each data record. An example of a data file for this analysis would look like this:

```
8592    (record 1 - Pretest = 85 & Posttest = 92)
7689    (record 2 - Pretest = 76 & Posttest = 89)
5276    (record 3 - Pretest = 52 & Posttest = 76)
```

The syntax of the command to perform one or more matched pairs t-tests is:

**TTEST <Variable list> WITH <Variable list>**

The keyword WITH is mandatory if a variable list is specified (i.e., more than one t-test is requested). If only one t-test is being requested, the keyword WITH may be omitted. In our pretest-posttest example, the commands would be:

```
TTEST PRETEST WITH POSTTEST
TTEST PRETEST POSTTEST
TT V1 WITH V2    (TTEST may be abbreviated as TT)
```

In a matched pairs t-test, it does not matter which variable is listed first in the command. Identical results would be obtained with the command:

### TTEST POSTTEST WITH PRETEST

When a variable list is specified as part of the TTEST command, more than one t-test analysis will be performed. For example, the following command will perform four t-test analyses (V1 with V9, V1 with V23, V7 with V9, and V7 with V23):

### TTEST V1 V7 WITH V9 V23

### Example of a t Test for Matched Pairs Printout

StatPac For Windows						
<u>Effect of Experimental Drug LN-143 on Reaction Time</u>						
	N	Mean	SD	t	DF	p
Reaction Time Immediately Before Ingestion	10	40.700	13.022	2.694	9	.025
Reaction Time 1 Hour Later	10	29.500	13.243			
Correlation Coefficient = .499						
Valid Cases = 10						
Missing Cases = 0						
Response Percent = 100.0%						

## T-Test For Independent Groups

The other kind of t-test is for independent groups and it is used for noncorrelated data. If each case in the data file is to be assigned to one group or the other based on another variable, use the t-test for independent groups. For example, to compare reading scores between males and females, split the reading scores into two groups depending upon whether the person is male or female. (Each record in the data file is assigned to one group or the other.)

M83 (record 1 - male with score of 83)  
 F91 (record 2 - female with score of 91)  
 F84 (record 3 - female with score of 84)

The syntax for the command to perform an independent groups t-test is:

TTEST <Var. list> WITH <Grouping var. list>=(<Code 1>)(<Code 2>)



As in the matched pairs t-test, the keyword WITH is only mandatory if a variable list is specified for the analyzed variable or the grouping variable. If only one t-test is requested, its use is optional.

In the above example, SCORE is the variable under analysis and SEX is the variable used to assign records to one group or the other. The commands to perform this t-test are:

**TTEST SCORE WITH SEX=(M)(F)**

**TTEST SCORE WITH SEX=(M/m)(F/f)**

In the second example, notice that both upper and lower case codes are specified; they are separated from each other by a slash. This is done just in case our data entry operators were not consistent in the way they entered the data. That is, sometimes a male was designated with an "M", and other times with an "m". If you are certain that upper case was always used, you could use the first command.

When a variable list is specified, several t-tests will be performed. For example, the following statement would request three different t-tests between males and females (one for each type of score):

**TTEST SCORE1 SCORE2 SCORE3 WITH SEX=(M)(F)**

There is no limit on the number of codes that can be specified as part of a group. For example, let's say an INCOME variable is coded into five income groups:

What is your annual income?

1=Under \$10,000

2=\$10,000 - \$20,000

3=\$21,000 - \$30,000

4=\$31,000 - \$40,000

5=Over \$40,000

To compare scores for those that make up to \$30,000 with those that make over \$30,000 per year, the command would be:

**TTEST SCORE WITH INCOME=(1/2/3)(4/5)**

When performing a t-test for independent groups, the program will accept a wide variety of user styles and formats. Two basic formats are possible. These are:

**(<Code>/<Code>) or (<Code>-<Code>)**

All of the following would be valid requests when entering the code(s) or value(s) to split the data into two groups. Notice that the reserved words LO and HI are valid when specifying a range of codes or values.

- (A/B/D) (Place all cases with codes A, B or D in this group)  
 (A-D) (Place all cases with codes A to D in this group)  
 (LO-D) (Place all cases with codes up to D in this group)  
 (6-9) (Place all cases with codes 6-9 in this group)  
 (LO-21) (Place all cases with up to a 21 in this group)  
 (22-HI) (Place all cases with a 22 or higher in this group)

### ***Example of a t-Test for Independent Groups Printout***

<b>StatPac For Windows</b>						
<b><u>A Comparison of Typing Speed Between Males and Females</u></b>						
Dependent Variable - Typing Speed (Words Per Minute). This is using an average word length of 5 characters per word.						
Variable Used To Group Cases - Gender						
	N	Mean	SD	t	DF	p
Male	8	36.000	14.243	5.218	13	.000
Female	7	72.000	12.179			
Valid Cases = 15						
Missing Cases = 0						
Response Percent = 100.0%						

### ***Non-Parametric Statistics***

The non-parametric equivalents of the t-test can be requested with the NP=Y option. Either the Wilcoxon test or the Mann-Whitney U test will be printed depending on whether you are performing a matched pairs or independent groups t-test.

The Wilcoxon test for correlated samples is the non-parametric equivalent of the matched pairs t-test. The data is assigned rank values and the differences between the ranks are computed. The Wilcoxon test statistic is the minimum of positive and negative differences in ranks. If the number of cases is greater than or equal to ten, the probability is calculated from the normal distribution. When there are fewer than ten cases, refer to the appendix to determine the probability.

### ***Example of Wilcoxon Statistic Printout***

<b><u>Non-Parametric Statistics</u></b>	
Number Of Positive Differences =	8
Sum Of The Positive Differences =	48.500
Number Of Negative Differences =	2
Sum Of The Negative Differences =	6.500
Number Of Non-Zero Differences =	10
Wilcoxon Test Statistic =	6.500

The Mann-Whitney U test is the non-parametric equivalent of the t-test for independent groups. It may be used to evaluate the difference between two population distributions. The data is first ranked. The Mann-Whitney U is the number of times that one group is smaller than the other.

For sample sizes of less than twenty, refer to the appendix to find the probability of U. If the sample size is twenty or more, the distribution approximates the normal distribution, and the normal deviate will be used to calculate the probability. The Mann-Whitney U may be selected by using MW=Y or suppressed by using MW=N.

### ***Example of Mann-Whitney U Statistic Printout***

<u>Non-Parametric Statistics</u>	
Median Group 1 =	36.500
Average Rank Group 1 =	4.500
Median Group 2 =	67.000
Average Rank Group 2 =	12.000
Mann-Whitney U =	56.000
Standard Normal Deviate =	3.240
Two-Tailed Probability =	.001

### ***Labeling and Spacing Options***

Option	Code	Function
Labeling	LB	Sets the column headings to print the variable label (LB=E), the variable name (LB=N), or the variable number (LB=C).
Labeling Width	LW	Set the maximum width (in inches) for the variable labels on the stub.
Exact Width	EW	When EW=Y, the labeling width for the stub will be exactly what is specified with the LW option. When EW=N, the width of the stub will self-adjust based on the length of the stub labels.
Column Spacing	CS	Sets the spacing (in inches) between the columns.
Decimal Places	DP	Sets the number of decimal digits that will be shown.
Print Percent Symbol	PP	Sets whether the percent symbol is shown.
Label Justification	LJ	Sets the justification for the banner labels.
Blank Lines Between Rows	BL	Sets the number of blank lines between each variable on the stub when OA=Y.

---

## **CORRELATE Command**

Correlation is a measure of association between two variables. A correlation coefficient can be calculated for ordinal, interval or ratio-type data. This program can print descriptive statistics and a correlation matrix for up to 88 variables.

The syntax of the command to run a correlation analysis is:

**CORRELATE <Variable list>**

For example, to run a simple correlation of EDUCATION and INCOME, you could type the command as:

```

CORRELATE EDUCATION INCOME
CORRELATE V1, V2
CO V1 V2      (CORRELATE can be abbreviates as CO)

```

The correlation program can also correlate more than two variables. For example, to print a correlation matrix of AGE, INCOME, ASSETS and TEST-SCORE, you would type the command:

```

CORRELATE AGE, INCOME, ASSETS, TEST-SCORE

```

The output would contain a correlation matrix of all possible combinations of variable pairs. Several statistics can be printed for each pair of variables. These are the correlation coefficient, number of valid records, standard error of the estimate, t statistic and probability of t.

### ***Type of Correlation Coefficient***

StatPac can calculate two different kinds of correlation coefficients: Spearman's rank-difference correlation coefficient and Pearson's product-moment correlation coefficient. When calculating a correlation coefficient for ordinal data, choose Spearman's rank-difference technique. For interval or ratio-type data, select Pearson's product-moment formula.

It is your responsibility to select the appropriate type of statistic. This can be accomplished by using the TY option. The TY option may be specified as S (Spearman's) or P (Pearson's). For example, when analyzing interval-type variables, type:

```

OPTIONS TY=P

```

### ***Descriptive Statistics***

Descriptive statistics can be selected or rejected with the option DS=Y or DS=N. If Pearson's product-moment correlation is selected, the output will include the number of records, mean and standard deviation. Only the number of records will be printed if Spearman's rank-difference correlation is selected.

### ***Example of a Descriptive Statistics Printout***

Variables in the Analysis - Descriptive Statistics				
Variable		N	Mean	SD
V1	Gross Income (After Expenses)	15	129.936	6.672
V2	Number Of Stores	15	454.000	4.472
V3	Advertising Space Budget	15	58290.267	3147.856
V4	Number Of Salesmen In The Sales Force	15	261.733	71.682
V5	Sales Volume (Dollar Volume Amount)	15	3139.067	940.825
V6	Number Of Units Sold	15	94.267	23.020

## ***Simple Correlation Matrix***

The correlation matrix may be printed or suppressed with the SC=Y or SC=N option respectively. Most of the time, you'll probably want to print the correlation matrix. However, there may be times when you only want descriptive statistics and/or Cronbach's alpha reliability statistic.

## ***Correlation Coefficient***

The correlation coefficient(s) can be printed with the CC option. The option CC=Y will print the correlation coefficient while CC=N will suppress it.

The value of a correlation coefficient can vary from minus one to plus one. A minus one indicates a perfect negative correlation, while a plus one indicates a perfect positive correlation. A correlation of zero means there is no relationship between the two variables.

## ***Number Of Cases***

The number of cases (records) used to calculate the correlation coefficient can be printed with NC=Y. This may or may not be the same as the number of records in the data file. If either the X or Y value is missing from a pair of data, the record will be skipped and not included in the analysis.

## ***Standard Error***

The standard error of the estimate for a correlation coefficient measures the standard deviation of the data points as they are distributed around the regression line. The standard error of the estimate can be used to specify the limits and confidence interval for a correlation coefficient. It can only be calculated for interval or ratio-type data. The standard error can be printed using the option SE=Y.

## ***T Statistic***

The significance of the correlation coefficient is determined from the student's t statistic. The formula to calculate the t statistic depends upon which type of correlation coefficient is specified. The t statistic can be printed or not by using the option TT=Y or TT=N, respectively. Although StatPac does not calculate the F statistic, it is simply the square of the t statistic.

## ***Probability Of Chance***

The probability of the t statistic indicates whether the observed correlation coefficient occurred by chance if the true correlation is zero. It can be printed with the option PR=Y. StatPac uses a two-tailed test to derive this probability from the t distribution. Probabilities of .05 or less are generally considered significant, implying that there is a relationship between the two variables.

When the t statistic is calculated for Spearman's rank-difference correlation coefficient, there must be at least 30 cases before the t distribution can be used to determine the probability. If there are fewer than 30 cases, use the table in the appendix to find the probability of the correlation coefficient.

## Example of a Correlation Matrix Printout

Pearson's Product-Moment Correlation Matrix						
		V1	V2	V3	V4	V5
V1	r	.944				
	p	.000				
V2	r	.937	.994			
	p	.000	.000			
V3	r	.443	.422	.372		
	p	.082	.117	.173		
V4	r	.440	.447	.475	-.202	
	p	.083	.009	.004	.470	
V5	r	.982	.993	.990	.454	.579
	p	.000	.000	.000	.089	.024

## Cronbach's Alpha

Cronbach's alpha is a measure of the internal consistency of a group of items. It provides a unique estimate of reliability for a given test administration. The value of Cronbach's alpha may vary between zero and one. In general, it is a lower bound to the reliability of a scale of items. In other words, Cronbach's alpha tends to be a very conservative measure of reliability.

As well as being a measure of the reliability of a scale of items, Cronbach's alpha may also be interpreted as an estimate of the correlation of one test with an alternative form containing the same number of items.

## Labeling and Spacing Options

Option	Code	Function
Labeling	LB	Sets the labeling for descriptive statistics to print the variable label (LB=E), the variable name (LB=N), or the variable number (LB=C).
Column Width	CW	Sets the minimum width of the columns (in inches).
Column Spacing	CS	Sets the spacing (in inches) between the columns.
Decimal Places	DP	Sets the number of decimal digits that will be shown.

# Advanced Analyses

---

## Advanced Analyses Index

The Advanced Analyses module adds multivariate procedures that are not available in the basic StatPac for Windows package. These commands may be used in a procedure when the Advanced Analyses module has been installed.

ANOVA	Canonical	Cluster	Discriminant
Factor	Logit	Map	PCA
Probit	Regress	Stepwise	

---

## REGRESS Command

The REGRESS command may be used to perform ordinary least squares regression and curve fitting. Ordinary least squares regression (also called simple regression) is used to examine the relationship between one independent and one dependent variable. After performing an analysis, the regression statistics can be used to predict the dependent variable when the independent variable is known.

People use regression on an intuitive level every day. In business, a well-dressed man is thought to be financially successful. A mother knows that more sugar in her children's diet results in higher energy levels. The ease of waking up in the morning often depends on how late you went to bed the night before.

Quantitative regression adds precision by developing a mathematical formula that can be used for predictive purposes.

The syntax of the command to run a simple regression analysis is:

**REGRESS** <Dependent variable> <Independent variable>

- or -

**REGRESS** <Dependent variable list> **WITH** <Independent variable list>

For example, a medical researcher might want to use body weight (V1=WEIGHT) to predict the most appropriate dose for a new drug (V2=DOSE). The command to run the regression would be specified in several ways:

```
REGRESS DOSE WITH WEIGHT
REGRESS DOSE WEIGHT
RE V2 WITH V1    (Note: REGRESS may be abbreviated as RE)
RE V2 V1
```

Notice that the keyword WITH is an optional part of the syntax. However, if you specify a variable list for either the dependent or independent variable, the use of the WITH keyword is mandatory. When a variable list is specified, a separate regression will be performed for each combination of dependent and independent variables.

The purpose of running the regression is to find a formula that fits the relationship between the two variables. Then you can use that formula to predict values for the dependent variable when only the independent variable is known.

The general formula for the linear regression equation is:

$$y = a + bx$$

where:

- x is the independent variable
- y is the dependent variable
- a is the intercept
- b is the slope

## ***Curve Fitting***

Frequently, the relationship between the independent and dependent variable is not linear. Classical examples include the traditional sales curve, learning curve and population growth curve. In each case, linear (straight-line) regression would present a distorted picture of the actual relationship.

Several classical non-linear curves are built into StatPac and you can simply ask the program to find the best one. Transformations are used to find an equation that will make the relationship between the variables linear. A linear least squares regression can then be performed on the transformed data.

The process of finding the best transformation is known as "curve fitting". Basically, it is an attempt to find a transformation that can be made to the dependent and/or independent variable so that a least squares regression will fit the transformed data. This can be expressed by the equation:

$$(\text{transformed } y) = \text{intercept} + \text{slope} * (\text{transformed } x)$$

Notice the similarity to the least squares equation. The difference is that we are transforming the independent variable and predicting a transformed dependent variable. To solve for y, use the formula to untransform y and apply it to both sides of the equation.



$$y = \text{Untransformation of (intercept + slope * (transformed x))}$$

In addition to the built-in transformations, any non-linear relationship that can be expressed as a mathematical formula can be explored with the COMPUTE statement. It is possible to transform both the dependent and independent variables with the COMPUTE statement.

The transformations that are built into StatPac are known as Box-Cox transformations. They are:

Options	Transformation
TX=A or TY=A	Automatic
TX=B or TY=B	Reciprocal
TX=C or TY=C	Reciprocal root
TX=D or TY=D	Reciprocal fourth root
TX=E or TY=E	Log
TX=F or TY=F	Fourth root
TX=G or TY=G	Square root
TX=H or TY=H	No transformation
TX=I or TY=I	Robust technique (no transformation)

For example, to apply a square root transformation to the independent variable and no transformation to the dependent variable, the options statement would be:

**OPTIONS TX=G TY=H**

The following option statement would try to fit your data to a classical S-Curve. It says to apply a reciprocal transformation to the independent variable and a log transformation to the dependent variable:

**OPTIONS TX=B TY=E**

The program also contains an automatic feature to search for the best transformation. When TX or TY is set to automatic, the program will select the transformation that produces the highest r-squared. To get a complete table of all combinations of the transformations, set both TX and TY to automatic.

**OPTIONS TX=A TY=A**

The result will produce a table of all possible combinations of transformations and the R-Squared statistics:

## Example of a Transformation Table

<u>Table of Transformations and R-Squared Statistics</u>							
<u>Transformation To IV</u>							
Transformation To DV	Recip.	Recip. Root	Recip. Fourth Root	Log	Fourth Root	Square Root	None
Reciprocal	.2279	3075	3661	.4253	.4684	.4855	.4599
Recip. Root	.2244	3048	3643	.4250	.4700	.4889	.4659
1/(4th Root)	.2225	3032	3630	.4244	.4703	.4901	.4685
Log	.2204	3013	3615	.4236	.4703	.4911	.4708
Fourth Root	.2182	2993	3598	.4225	.4701	.4917	.4728
Square Root	.2159	2971	3579	.4211	.4695	.4920	.4746
None	.2109	2921	3535	.4177	.4676	.4919	.4774
Transformations Selected: IV=Square Root DV=Square Root							

The Box-Cox transformations can be expressed mathematically:

	Transformation	Untransformation
Reciprocal	$z = 1/(y+k)$	$y = (z)^{-1} - k$
Reciprocal root	$z = 1/(y+k)^{.5}$	$y = (z)^{-2} - k$
Reciprocal fourth root	$z = 1/(y+k)^{.25}$	$y = (z)^{-4} - k$
Log	$z = \text{Log}(y+k)$	$y = \text{Exp}(z) - k$
Fourth root	$z = (y+k)^{.25}$	$y = (z)^4 - k$
Square root	$z = (y+k)^{.5}$	$y = (z)^2 - k$

Notes:

- z is the transformed data value
- y is the original data value
- k is a constant used in the transformation

Two cautions should be noted when using transformations.

1. When a reciprocal transformation is used, the sign of the correlation coefficient may no longer indicate the direction of the relationship in the untransformed data.
2. Some transformations may not be possible for some data. For example, it is not possible to take the log or square root of a negative number or the reciprocal of zero. When necessary, StatPac will automatically add a constant to the data to prevent this type of error.

One problem with least squares regression is its susceptibility to extreme or unusual data values. In many cases, even a single extreme data value can distort the regression results. A technique called robust regression is included in StatPac to overcome this problem. Robust regression mathematically adjusts extreme data values through an iterative process. The effect is to reduce the distortion in the regression line caused by the outlying data value(s).

The robust process makes successive adjustments to extreme data values by examining the median residual and using a weighted least squares regression to adjust the outliers.

If robust regression is specified for either the x or y transformation, no other built-in transformations will be used (even if a transformation is specified for the other variable).

## Statistics

The regression statistics provide the information needed to adequately evaluate the "goodness-of-fit"; that is, how well the regression line explains the actual data. The statistics include correlation information, descriptive statistics, error measures and regression coefficients. This data can be used to predict future values for the dependent variable and to develop confidence intervals for the predictions.

### Example of a Statistics Printout

Goodness Of Fit

Mean of Residuals = 0.802  
Standard Deviation of Residuals = 45.945  
Mean Absolute Percent Error = 5.493%  
Mean Percent Error = -0.384%  
Mean Square Error = 2041.199

Regression Statistics

Valid Cases = 30  
Missing Cases = 0  
Correlation Coefficient = 0.701  
Degrees of Freedom = 28  
Adjusted R-Squared = 0.474  
Standard Error of the Estimate = 0.927

Regression Coefficients Table

	Coefficient	Estd. Std. Error	T-Value	Significance
Intercept	22.585	0.580	38.954	0.0000
Slope	2.204	0.423	5.208	0.0000

Correlation is a measure of association between two variables. StatPac calculates the Pearson product-moment correlation coefficient. Its value may vary from minus one to plus one. A minus one indicates a perfect negative correlation, while a plus one indicates a perfect positive correlation. A correlation of zero means there is no relationship between the two variables. When a transformation has been specified, the correlation coefficient refers to the relationship of the transformed data.

The coefficient of determination (r-squared) is the square of the correlation coefficient. Its value may vary from zero to one. It has the advantage over the correlation coefficient in that it may be interpreted directly as the proportion of variance in the dependent variable that can be accounted for by the regression equation. For example, an r-squared value of .49 means that 49% of the variance in the dependent variable can be explained by the regression equation. The other 51% is unexplained error.

The standard error of estimate for regression measures the amount of variability in the points around the regression line. It is the standard deviation of the data points as they are distributed around the regression line. The standard error of the estimate can be used to specify the limits and confidence of any prediction made and is useful to obtain confidence intervals for  $y'$  given a fixed value of  $x$ .

Regression analysis enables us to predict one variable if the other is known. The regression line (known as the "least squares line") is a plot of the expected value of the dependent variable for all values of the independent variable.

The difference between the observed and expected value is called the residual. It can be used to calculate various measures of error. The measures of error in StatPac are the mean percent error (MPE), mean absolute percent error (MAPE) and the mean squared error (MSE).

Using the regression equation, the variable on the  $y$  axis may be predicted from the score on the  $x$  axis. The slope of the regression line ( $b$ ) is defined as the rise divided by the run. The  $y$  intercept ( $a$ ) is the point on the  $y$  axis where the regression line would intercept the  $y$  axis. The slope and  $y$  intercept are incorporated into the regression equation as:  $y = a + bx$

The significance of the slope of the regression line is determined from the student's  $t$  statistic. It is the probability that the observed correlation coefficient occurred by chance if the true correlation is zero. StatPac uses a two-tailed test to derive this probability from the  $t$  distribution. Probabilities of .05 or less are generally considered significant, implying that there is a relationship between the two variables. Although StatPac does not calculate the  $F$  statistic, it is simply the square of the  $t$  statistic for the slope.

## Data Table

The data table provides a detailed method to examine the errors between the predicted and actual values of the dependent variable. StatPac allows printing of the table to more closely study the residuals. Typing the option  $DT=Y$  will cause the output to include a data table.

## Example of a Data Table

<u>Table of Observed Data, Predicted Data, and Error</u>								
<u>Rec</u>	<u>IV</u>	<u>DV</u>	<u>Predicted</u>	<u>Residual</u>	<u>Std. Resid.</u>	<u>% Error</u>	<u>- 95% CI</u>	<u>+ 95% CI</u>
1	13	510	629.9	-119.9	-2.610	-23.5	611.1	648.9
2	0.1	533	542.0	-9.0	-0.197	-1.7	499.6	586.1
3	0.8	558	603.0	-45.0	-0.979	-8.1	578.7	627.8
4	1.1	569	619.8	-50.8	-1.106	-8.9	599.4	640.6
5	0.1	580	542.0	38.0	0.826	6.5	499.6	586.1
6	13	581	629.9	-48.9	-1.064	-8.4	611.1	648.9
7	1.5	603	639.3	-36.3	-0.790	-6.0	621.5	657.3
8	13	610	629.9	-19.9	-0.433	-3.3	611.1	648.9
9	0.9	612	608.9	3.1	0.068	0.5	586.0	632.2
10	3.0	618	697.1	-79.1	-1.721	-12.8	670.7	723.9
11	2.9	630	693.7	-63.7	-1.386	-10.1	668.3	719.5
12	1.1	633	619.8	13.2	0.287	2.1	599.4	640.6
13	1.1	643	619.8	23.2	0.504	3.6	599.4	640.6
14	0.9	645	608.9	36.1	0.786	5.6	586.0	632.2

## ***Outlier Definition and Adjustment***

Outliers (extreme data points) can have dramatic effects on the slope of the regression line. One method to deal with outliers is to use robust regression (TX=I or TY=I). There are two other common methods to deal with outliers. The first is to simply eliminate any records that contain an outlier and then rerun the regression without those data records. The other method is known as *data trimming*, where the highest and lowest extreme values are replaced with a value that limits the standardized residual to a predetermined value. The OA option is used to set the outlier adjustment method. It may be set to OA=N (none), OA=D (delete), or OA=A (adjust).

Both methods use a two-step process. First the regression is performed using the actual values for the dependent variable and standardized residuals are calculated for each predicted value. When a standardized residual exceeds a given z-value, the record is flagged. Then the regression is run again and the flagged records are either eliminated (OA=D), or the value of the dependent variable is adjusted to the value defined by the outlier definition z-value (OA=A). For example, if the outlier definition is set to 1.96 standard deviations (OD=1.96), the upper and lower two and a half percent of the outliers would be flagged. Then the dependent variables for the flagged records would be modified to a value that would produce an outlier of plus or minus 1.96. Finally, the regression would be rerun using the modified dependent variable values for the flagged records. Flagged data records will be shown with an asterisk in the data table.

It is important to note that the outlier adjustment process is only performed once because each regression would produce a new set of standardized residuals that would exceed the outlier definition value (OD=z). That is, any set of data with sufficient sample size will yield a set of outliers, even if the data has already been adjusted. Allowing the outlier adjustment process to repeat indefinitely would eventually result in an adjustment to nearly every data record.

When outlier adjustment is used, the program will also report adjusted means and standard deviations for the dependent variable. This refers to the recalculated mean after deleting or adjusting the data.

## ***Confidence Intervals & Confidence Level***

Confidence intervals provide an estimate of variability around the regression line. Narrow confidence intervals indicate less variability around the regression line. The option CI=Y will include the confidence intervals in the data table.

Prediction intervals, rather than confidence intervals, should be used if you intend to use the regression information to predict new values for the dependent variable. Both the confidence intervals and the prediction intervals are centered on the regression line, but the prediction intervals are much wider. The option CI=P will print the prediction intervals in the data table.

The actual confidence or prediction interval is set with the CL option. The CL option specifies the percentage level of the interval. For example, if CI=P and CL=95, the 95% prediction intervals would be printed in the data table.

## ***Residual Autocorrelation Function Table***

Examining the autocorrelation of the residuals is often used in time-series analysis to evaluate how well the regression worked. It is a way of looking at the "goodness-of-fit" of the regression line. If the residuals contain a pattern, the regression did not do as well as we might have desired.

A residual autocorrelation table is the correlation between values that occur at various time lags. For example, at time lag one, you are looking at the correlation between adjacent values; at time lag two, you are looking at the correlation between every other value, etc. To select the residual autocorrelation function table, type the option AC=Y.

### ***Example of a Residual Autocorrelation Function Table***

<u>Residual Autocorrelation Function Table</u>				
<u>Lag</u>	<u>Value</u>	<u>T-Value</u>	<u>1 SE</u>	<u>2 SE</u>
1	.254	1.389	.183	.365
2	.202	1.043	.194	.388
3	.318	1.584	.201	.402
4	.178	.822	.217	.434
5	.286	1.291	.222	.444
6	.124	.529	.234	.468
7	.258	1.095	.236	.472
8	.246	1.004	.245	.491
9	.392	1.547	.253	.507
10	.223	.816	.273	.546
Box-Pierce Chi-Square Statistic (DF=10) = 19.991				
Probability = 0.029				

### ***Expanding Standard Error***

You may use the EX option to set the standard error limits of the residual autocorrelation function to a fixed value or to expand with increasing time lags.

A study of sampling distributions on autocorrelated time series was made by Bartlett in 1946. He found that, as one goes out further in time, the standard error increases with successive time lags. ("Theoretical Specifications of Sampling Properties of Autocorrelated Time Series", Bartlett, 1946.) It is only in recent years that his findings have been accepted by the forecasting community.

When EX=Y, the residual autocorrelation function error limits will widen with each successive time lag. If EX=N, the standard error limits will remain constant.

### ***Force Constant to Zero***

The option CZ=Y can be used to calculate a regression equation with the constant equal to zero. If this is done, the regression line is forced through the origin. Note that forcing the constant to zero disables calculation of the correlation coefficient, r-squared, and the standard error of estimate. For this reason, it is not possible to set the transformation parameter for either the independent or dependent variable to automatic, because there are no r-squared statistics to compare. Furthermore, confidence intervals, which are calculated from the standard error of estimate, cannot be computed. The option CZ=N results in a standard regression equation.

### ***Save Results***

Many times researchers want to save results for future study. By using the option SR=Y, the predicted values, residuals and confidence or prediction intervals can be

saved so they can be merged with the original data file. At the completion of the analysis, you will be given the opportunity to merge the predictions and residuals.

### ***Predict Interactively***

When performing a regression, predicting values for the dependent variable for specific values of the independent variable may be desired. This is known as interactive prediction. Select interactive prediction by entering the option PR=Y. After the completion of the tabular outputs, the user will be prompted to enter a value for the independent variable. The program will predict the value for the dependent variable based on the regression equation. Confidence and/or prediction intervals will also be given.

### ***Labeling and Spacing Options***

Option	Code	Function
Labeling	LB	Sets the labeling for descriptive statistics to print the variable label (LB=E), the variable name (LB=N), or the variable number (LB=C).
Decimal Places	DP	Sets the number of decimal digits that will be shown.

---

## **STEPWISE Command**

Multiple regression is an extension of simple regression. It examines the relationship between a dependent variable and two or more explanatory variables (also called independent or predictor variables). Multiple regression is used to:

1. Predict the value of a dependent variable using some or all of the independent variables. The aim is generally to explain the dependent variable accurately with as few independent variables as possible.
2. To examine the influence and relative importance of each independent variable on the dependent variable. This involves looking at the magnitude and sign of the standardized regression coefficients as well as the significance of the individual regression coefficients.

The syntax of the command to run a stepwise regression is:

**STEPWISE <Dependent variable> <Independent variable list>**

For example, we might try to predict annual income (V1=INCOME) from age (V2=AGE), number of years of school (V3=SCHOOL), and IQ score (V4=IQ). The command to run the regression could be specified in several different ways:

**STEPWISE INCOME, AGE, SCHOOL, IQ**

**STEPWISE V1,V2,V3,V4**

**ST INCOME V2-V4** (Note: STEPWISE may be abbreviated as ST)

**STEPWISE V1-V4**

In each example, the dependent variable was specified first, followed by the independent variable list. The variable list itself may contain up to 200 independent variables and can consist of variable names and/or variable numbers. Either a comma or a space can be used to separate the variables from each other.

The multiple regression equation is similar to the simple regression equation. The only difference is that there are several predictor variables and each one has its own regression coefficient.

The multiple regression equation is:

$$Y' = a + b_1 x_1 + b_2 x_2 + b_3 x_3 + \dots b_n x_n$$

where

$Y'$  is the predicted value

$A$  is a constant

$B_1$  is the estimated regression coefficient for variable 1

$X_1$  is the score for variable 1

$B_2$  is the estimated regression coefficient for variable 2

$X_2$  is the score for variable 2

## Descriptive Statistics

The mean and standard deviations for all the variables in the equation can be printed with the DS=Y option.

### Example of a Descriptive Statistics Printout

<u>Variables in the Analysis - Descriptive Statistics</u>			
Variable		Mean	SD
DV1	Employment	65.317	3.512
V2	Year	1954.500	4.761
V3	Total Population	117.424	6.956
V4	Size of the Armed Forces	260.669	69.592
V5	Unemployment	319.331	93.446
V6	Gross National Product	387.698	99.395
V7	Gross Nat'l Prod Deflator	101.681	10.792

## Regression Statistics

The regression statistics can be selected with option RS=Y. They give us an overall picture of how successful the regression was.

The coefficient of multiple determination, frequently referred to as *r-squared*, can be interpreted directly as the proportion of variance in the dependent variable that can be accounted for by the combination of predictor variables. A coefficient of multiple determination of .85 means that 85 percent of the variance in the dependent variable



can be explained by the combined effects of the independent variables; the remaining 15 percent would be unexplained.

The coefficient of multiple correlation is the square root of the coefficient of multiple determination. Its interpretation is similar to the simple correlation coefficient. It is basically a measure of association between the predicted value and the actual value.

The standard error of the multiple estimate provides an estimate of the standard deviation. It is used in conjunction with the inverted matrix to calculate confidence intervals and statistical tests of significance.

When there are fewer than 100 records, StatPac will apply an adjustment to the above three statistics, and the adjusted value will be printed. The adjustment is for a small n and its value should be used.

The variability of the dependent variable is made up of variation produced by the joint effects of the independent variables and some unexplained variance. The overall F-test is performed to determine the probability that the true coefficient of multiple determination is zero. Typically, a probability of .05 or less leads us to reject the hypothesis that the regression equation does not improve our ability to predict the dependent variable.

### ***Example of the Regression Statistics Printout***

<u>Regression Statistics</u>	
Coefficient of Multiple Determination = 0.995 ( Corrected = 0.992 )	
Coefficient of Multiple Correlation = 0.998 ( Corrected = 0.996 )	
Standard Error of the Multiple Estimate = 0.305 ( Corrected = 0.394 )	
F-Ratio = 330.285	
Degrees of Freedom = 6 & 9	
Probability of Chance = 0.000	
Number of Valid Cases = 16	
Number of Missing Cases = 0	
Response Percent = 100.0 %	

### ***Regression Coefficients***

The regression coefficients can be printed with the RC=Y option. The output includes the constant, coefficient, beta weight, F-ratio, probability, and standard error for each independent variable.

Each coefficient provides an estimate of the effect of that variable (in the units of the raw score) for predicting the dependent variable. The beta weights, on the other hand, are the standardized regression coefficients and represent the relative importance of each independent variable in predicting the dependent variable.

The F-ratio allows us to calculate the probability that the influence of the predictor variable occurred by chance. The t-statistic for each independent variable is equal to the square root of its F-ratio.

The standard error of the *i*th regression coefficient can be used to obtain confidence intervals about each regression coefficient in conjunction with its F-ratio.

### Example of Regression Coefficients Printout

<u>Regression Coefficients Table</u>					
Var.	Coeff.	Beta	F-Ratio	Prob.	Std. Error
V2	1.829	2.480	16.127	0.003	0.455
V3	-0.051	-0.101	0.051	0.826	0.226
V4	-0.010	-0.205	23.252	0.001	0.002
V5	-0.020	-0.538	17.110	0.003	0.005
V6	-0.036	-1.014	1.144	0.313	0.033
V7	0.015	0.046	0.031	0.863	0.085
Const.	-3482.259		12093.567	0.000	31.665

### Simple Correlation Matrix

After performing a regression analysis, it is a good idea to review the simple correlation matrix (SC=Y). If two variables are highly correlated, it is possible that the matrix is not well conditioned and it might be beneficial to run the regression again without one of the variables. If the coefficient of multiple determination does not show a significant change, you might want to leave the variable out of the equation.

### Example of a Simple Correlation Printout

<u>Simple Correlation Matrix</u>							
		DV1	V2	V3	V4	V5	V6
V2	r	0.971					
	p	0.000					
V3	r	0.960	0.994				
	p	0.000	0.000				
V4	r	0.457	0.417	0.364			
	p	0.075	0.108	0.165			
V5	r	0.502	0.668	0.687	-0.177		
	p	0.047	0.005	0.003	0.511		
V6	r	0.984	0.995	0.991	0.446	0.604	
	p	0.000	0.000	0.000	0.083	0.013	
V7	r	0.971	0.991	0.979	0.465	0.621	0.992
	p	0.000	0.000	0.000	0.070	0.010	0.000

### Partial Correlation Matrix

The partial correlation matrix (often called the variance-covariance matrix) is obtained from the inverse of simple correlation matrix. It can be selected with the

option PC=Y. This is useful in studying the correlation between two variables while holding all the other variables constant.

A significant partial correlation between variables A and B would be interpreted as follows: When all other variables are held constant, there is a significant relationship between A and B. The partial correlation matrix will be printed for those variables remaining in the equation after the stepwise procedure.

### ***Example of a Partial Correlation Matrix Printout***

<u>Partial Correlation Matrix</u>					
	V2	V3	V4	V5	V6
V3	-0.388				
V4	0.549	0.189			
V5	0.824	0.758	-0.619		
V6	0.802	0.833	-0.469	-0.946	
V7	-0.186	-0.659	0.349	0.555	0.649

### ***Inverted Correlation Matrix***

The solution to a multiple regression problem is obtained through a technique known as matrix inversion. The inverted correlation matrix is the inversion of the simple correlation matrix. It may be selected with the option IC=Y.

In examining the inverted matrix, we are specifically interested in the values along the diagonal. They provide a measure of how successful the matrix inversion was. If all the values on the diagonal are close to one, the inversion was very successful and we say the matrix is "well conditioned". If, however, we have one or more diagonal values that are high (greater than ten), we have a problem with collinearity (high correlations between independent variables).

### Example of an Inverted Matrix Printout

<u>Inverse of Simple Correlation Matrix</u>						
	V2	V3	V4	V5	V6	V7
V2	758.981					
V3	213.646	399.151				
V4	-28.672	-7.150	3.589			
V5	-131.640	-87.837	6.795	33.619		
V6	-934.035	-703.991	37.544	231.872	1788.513	
V7	59.747	153.318	-7.693	-37.463	-319.737	135.532

### Print Each Step

You can print the statistics for each step of the stepwise procedure using the option PS=Y. This may be important when you want to study how the inclusion or deletion of a variable affects the other variables.

### Example of the Print Steps Output

<u>Step1</u>	Variable V6 Entered	R-Squared =0.967 F(1,14)=415.103, p=0.000			
	<u>Var.</u>	<u>Coeff.</u>	<u>Beta</u>	<u>F-Ratio</u>	<u>Prob.</u>
	V6	0.035	0.984	415.103	0.000
	Constant	51.844			
<u>Step2</u>	Variable V5 Entered	R-Squared =0.981 F(2,13)=329.498, p=0.000			
	<u>Var.</u>	<u>Coeff.</u>	<u>Beta</u>	<u>F-Ratio</u>	<u>Prob.</u>
	V5	-0.005	-0.145	8.925	0.010
	V6	0.038	1.071	489.314	0.000
	Constant	52.382			
No other variables met the significance level for entry or removal					

### Summary Table

A good way to get an overview of how the steps proceeded and what effect each step had upon the r-squared is to print a summary table. To print the summary table, use the option ST=Y.

### Example of a Summary Table

<u>Stepwise Regression Summary Table</u>						
Step No.	Variable		Multiple		Increase	No. of IVs
	Entered	Removed	R	RSQ	In RSQ	
1	V6		0.984	0.967	0.967	1
2	V5		0.990	0.981	0.013	2

### Data Table

The data table provides a detailed method to examine the residuals. StatPac allows printing of the table to more closely study the residuals. Using the option DT=Y will cause the output to include a data table.

A "residual" is the difference between the observed value and the predicted value for the dependent variable (the error in the prediction). The standardized residuals which appear in the data table are the residuals divided by the standard error of the multiple estimate. Therefore, the standardized residuals are in standard deviation units. In large samples, we would expect 95 percent of the standardized residuals to lie between -1.96 and 1.96.

### Example of a Data Table

<u>Table of Observed Data, Predicted Data, and Error</u>							
Rec	Observed	Predicted	Residual	Std. Resid	% Error	- 95% CI	+ 95% CI
1	60.323	60.056	0.267	0.877	0.4	59.684	60.427
2	61.122	61.216	-0.094	-0.308	-0.2	60.867	61.565
3	60.171	60.125	0.046	0.152	0.1	59.740	60.509
4	61.187	61.597	-0.410	-1.345	-0.7	61.268	61.926
5	63.221	62.911	0.310	1.016	0.5	62.616	63.207
6	63.639	63.888	-0.249	-0.818	-0.4	63.509	64.268
7	64.989	65.153	-0.164	-0.538	-0.3	64.780	65.526
8	63.761	63.774	-0.013	-0.043	0.0	63.480	64.068
9	66.019	66.005	0.014	0.047	0.0	65.793	66.216
10	67.857	67.402	0.455	1.494	0.7	67.192	67.611
11	68.169	68.186	-0.017	-0.057	0.0	67.959	68.414
12	66.513	66.552	-0.039	-0.128	-0.1	66.242	66.862
13	68.655	68.811	-0.156	-0.510	-0.2	68.550	69.071
14	69.564	69.650	-0.086	-0.281	-0.1	69.357	69.942
15	69.331	68.989	0.342	1.122	0.5	68.644	69.334
16	70.551	70.758	-0.207	-0.678	-0.3	70.422	71.094

### Outlier Definition and Adjustment

Outliers (extreme data points) can have a dramatic effect on the stability of a multiple regression model. There are two common methods to deal with outliers in multiple regression models. The first is to simply eliminate any records that contain an outlier and then rerun the regression without those data records. When using OA=D (setting the outlier adjustment to delete), records containing the highest and lowest extreme residuals are deleted from the analysis. The other method is where the dependent variable is adjusted for records with the highest and lowest extreme residuals. That is, the dependent variable is modified to a value that limits the standardized residual to a predetermined value. The OA option is used to set the

outlier adjustment method. It may be set to OA=N (none), OA=D (delete), or OA=A (adjust).

Both methods use a two-step process. First the regression is performed using the actual values for the dependent variable and standardized residuals are calculated for each predicted value. When a standardized residual exceeds a given z-value, the record is flagged. Then the regression is run again and the flagged records are either eliminated (OA=D), or the value of the dependent variable is adjusted to the value defined by the outlier definition z-value (OA=A). For example, if the outlier definition is set to 1.96 standard deviations (OD=1.96), the upper and lower two and a half percent of the outliers would be flagged. Then the dependent variables for the flagged records would be modified to a value that would produce an outlier of plus or minus 1.96. Finally, the regression would be rerun using the modified dependent variable values for the flagged records. Flagged data records will be shown with an asterisk in the data table.

The stepwise procedure presents a problem for data trimming. The stepwise procedure often reduces the number of independent variables to a subset of the original list of independent variables. Data trimming involves using the standardized residuals to adjust the value of the dependent variable for some records. Rerunning the stepwise procedure with different values for some of the dependent variables could result in a different set of independent variables being stepped into the model, especially when there are highly correlated independent variables. To avoid this problem, StatPac reruns the multiple regression using the same independent variables selected in the first stepwise procedure. These variables are forced into the model so that the analysis runs in the non-stepwise mode.

It is important to note that the outlier adjustment process is only performed once because each regression would produce a new set of standardized residuals that would exceed the outlier definition value (OD=z). That is, any set of data with sufficient sample size will yield a set of outliers, even if the data has already been adjusted. Allowing the outlier adjustment process to repeat indefinitely would eventually result in an adjustment to nearly every data record.

It is suggested that the user actually examine records that are flagged as extreme outliers before allowing the program to make any adjustments. Outlier adjustments assume that the data for all independent variables is acceptable. A misspelled data value for an independent variable could result in an extreme prediction that gets flagged as an outlier. Therefore, visual inspection is the best way to guarantee the successful handling of outliers.

### ***Confidence Intervals & Confidence Level***

Confidence intervals provide an estimate of variability around the regression line. Narrow confidence intervals indicate less variability around the regression line. The option CI=Y will include the confidence intervals in the data table.

Prediction intervals, instead of confidence intervals, should be used if you intend to use the regression information to predict new values for the dependent variable. Both the confidence intervals and the prediction intervals are centered on the regression line, but the prediction intervals are much wider. The option CI=P will print the prediction intervals in the data table.

The actual confidence or prediction interval is set with the CL option. The CL option specifies the percentage level of the interval. For example, if CI=P and CL=95, the 95% prediction intervals would be printed in the data table.

## ***Number of Variables to Force***

The ability to force variables into an equation is important for several reasons:

1. A researcher often wishes to replicate the analysis of another study and, therefore, to force certain core variables into the equation, letting stepwise regression choose from the remaining set.
2. Some variables may be cheaper or easier to measure, and the user may want to see whether the remaining variables add anything to the equation.
3. It is common to force certain design variables into the equation.
4. When independent variables are highly correlated, one of them may be more accurate than the rest, and you may want to force this variable into the equation.

The FO option specifies the number of variables to force into the regression equation. To perform a standard (non-stepwise) multiple regression, set the FO option to the number of independent variables or higher. FO=200 will always force all independent variables into the equation. Thus, the FO option may be used to eliminate the stepwise part of the multiple regression procedure. If you force all variables into the equation, the multiple regression will contain only one step, where all variables are included in the equation.

The variables to be forced are taken in order from the list of independent variables. For instance, the option FO=3 forces the first three variables from the list of independent variables. Therefore, any variables you want to force should be specified at the beginning of the independent variable list.

## ***F to Enter & F to Remove***

When faced with a large number of possible explanatory variables, two opposed criteria of selecting a regression equation are usually involved:

1. To make the equation useful for predictive purposes, we would like our model to include as many of the independent variables as possible so that reliable fitted values can be determined.
2. Because of the costs involved in obtaining information on a large number of independent variables, and subsequently monitoring them, we would like the equation to include as few of the independent variables as possible.

The compromise between these two extremes is generally called "selecting the best regression". This involves multiple executions of multiple regression in an attempt to add variables to improve prediction or remove variables to simplify the regression function. Stepwise regression provides a partial automation of this procedure.

An important property of the stepwise procedure is based on the fact that a variable may be indicated to be significant in an early stage, and, thus, be entered in the equation. After several other variables are added to the regression equation, however, the initial variable may be indicated to be insignificant. The combined effects of two or more independent variables capture the same variance as a variable entered early on in the stepwise process. This method is often referred to as forward inclusion with backward elimination.

The algorithm used by StatPac is as follows:

1. First, enter into the regression equation all variables which the user wishes to force into the equation.
2. Enter the predictor that produces the greatest decrease in the residual sum of squares from all remaining predictors whose entry is not inhibited by the F-to-enter.

3. Remove the predictor that makes the least increase in the residual sum of squares from all (non-forced) predictors whose removal is not inhibited by the F-to-remove inhibiting rule.

Note that step 2 is executed only when it is not possible to execute step 3. If neither can be executed, the stepping is complete.

The following should be considered when setting F-to-enter and F-to-remove values in the parameter table:

1. A variable is removed if the F-value associated with that variable is less than the F-to-remove value set in the parameter table. Similarly, a variable is added if the F-value associated with that variable would be greater than the F-to-enter value set in the parameter if that variable were entered in the current equation.
2. Care should be taken to ensure that the F-to-remove be less than the F-to-enter; otherwise, a variable would be entered and then removed at alternate steps.
3. The default values for the F-to-enter and F-to-remove for many mainframe packages and StatPac are 4.0 and 3.9, respectively, which provide useful starting values.
4. Forcing all variables in an equation will give the usual (non-stepwise) multiple regression results.
5. Setting the F-to-remove value low yields the forward inclusion method.
6. For the first run on a data set, it is common to set the F-to-enter and F-to-remove values low to execute a large number of steps.

### ***Residual Autocorrelation Function Table***

Examining the autocorrelation of the residuals is often used in time-series analysis to evaluate how well the regression worked. It is a way of looking at the "goodness-of-fit" of the regression line. If the residuals contain a pattern, the regression did not do as well as we might have desired.

A residual autocorrelation function table contains the correlation between values that occur at various time lags. For example, at time lag one, you are looking at the correlation between adjacent values; at time lag two, you are looking at the correlation between every other value, etc. To select the residual autocorrelation function plot, type the option AC=Y.

### ***Example of a Residual Autocorrelation Function Table***

<u>Residual Autocorrelation Function Table</u>				
<u>Lag</u>	<u>Value</u>	<u>T-Value</u>	<u>1 SE</u>	<u>2 SE</u>
1	-.346	-1.383	.250	.500
2	.081	.292	.278	.557
3	-.284	-1.015	.280	.560
4	-.064	-.214	.297	.594
5	.415	1.392	.298	.596

Number of Residuals = 16  
Standard Deviation of the Residuals = 0.222  
Durbin-Watson Statistic = 2.559  
Box-Pierce Chi-Square Statistic (DF=5) = 6.130  
Probability = 0.294



### ***Expanding Standard Error***

You may use the EX option to set the standard error limits of the residual autocorrelation function to a fixed value or to expand with increasing time lags.

A study of sampling distributions on autocorrelated time series was made by Bartlett in 1946. He found that, as one goes out further in time, the standard error increases with successive time lags. ("Theoretical Specifications of Sampling Properties of Autocorrelated Time Series", Bartlett, 1946.) It is only in recent years that his findings have been accepted by the forecasting community.

When EX=Y, the residual autocorrelation function error limits will widen with each successive time lag. If EX=N, the standard error limits will remain constant.

### ***Save Residuals***

Researchers often want to save the residuals in a file for further study. If further analysis of the residuals shows a pattern, the regression may not have captured all the variance it might have, and we may want to model the residuals to further explain the variance.

You can save the results in a file with the options command SR=Y. The dependent variables, predicted values, residuals and confidence or prediction intervals will be saved, and at the completion of the procedure you will be offered the opportunity to merge the saved data into the original data file. in the new file.

### ***Force Constant to Zero***

The option CZ=Y can be used to calculate a regression equation with the constant equal to zero. If this is done, the regression line is forced through the origin. Note that forcing the constant to zero disables calculation of the r-squared, coefficient of multiple correlation and the standard error of estimate. Furthermore, confidence intervals, which are calculated from the standard error of estimate, cannot be computed. The option CZ=N results in a standard regression equation.

### ***Mean Substitution***

Mean substitution is one method often used to reduce the problem of missing information. Often, multiple regression research is difficult because if one independent variable is not known, it is necessary to exclude the whole record from the analysis. It is possible that this can substantially reduce the number of records that are included in the analysis. Mean substitution overcomes this problem by replacing any missing independent variable with the mean of that variable. While this technique has the possibility of slightly distorting the results, it can make it possible to perform a regression with substantial missing data.

### ***Steps Limit***

The steps limit is simply the maximum number of steps that can occur. Each inclusion and deletion of a variable is counted as one step. The purpose of the steps limit is to limit computer time. The syntax for the steps limit option is SL=n, where n is the maximum number of steps allowed.

### ***Predict Interactively***

After performing a regression, you may want to predict values for the dependent variable. This is known as interactive prediction. Select interactive prediction by

entering the option PR=Y. You will then be prompted to enter a value for each independent variable, and the computer will use the regression coefficients to predict the dependent variable.

### ***Labeling and Spacing Options***

Option	Code	Function
Labeling	LB	Sets the labeling for descriptive statistics to print the variable label (LB=E), the variable name (LB=N), or the variable number (LB=C).
Column Width	CW	Sets the minimum width of the columns (in inches).
Column Spacing	CS	Sets the spacing (in inches) between the columns.
Decimal Places	DP	Sets the number of decimal digits that will be shown.

---

## **LOGIT and PROBIT Commands**

Probit and logistic regression analyses examine the relationship between a dichotomous dependent variable (takes on only two values) and one or more explanatory variables (also called independent or predictor variables). When the dichotomous variable (dependent variable) is coded as 0 or 1, its predicted value from probit or logistic regression is the estimated probability of it being 1.

Probit and logistic regressions are often used to answer yes/no type questions. For example: a banker wants to decide whether or not to make a loan, or a scientist wants to predict whether the rat lives or dies. Both questions are yes/no and could be coded as zero or one.

Deciding between probit or logistic regression is a matter of choice. In logistic regression, the estimated value of the dependent variable is based on the cumulative logistic distribution, and in probit regression it is based on the cumulative normal distribution. The logistic distribution is scarcely distinguishable from the cumulative normal distribution between response rates of .01 and .99, and therefore, the choice of probit or logistic regression is usually made on the basis of which technique the user is most familiar with.

The syntax of the commands and options to run probit or logistic regression are identical:

**PROBIT <Dependent variable> <Independent variable list>**

**LOGIT <Dependent variable> <Independent variable list>**

For example, a banker might want to predict successful loan repayment (V4=LOAN PAYBACK) from previous loan experience (V1=EXPERIENCE), number of credit cards (V2=CREDIT CARDS), and bank balance (V3=BALANCE). The command to run the regression could be specified in several ways.

**PROBIT LOAN PAYBACK, EXPERIENCE, CREDIT CARDS,  
BALANCE**

PROBIT V4, V1-V3 (Note: PROBIT may be abbreviated as PR)

LOGIT LOAN PAYBACK V1-V3

LO V4 V1 V2 V3 (Note: LOGIT may be abbreviated as LO)

In each example, the dependent variable is specified first, followed by the independent variable list. The variable list may contain up to 200 independent variables and can consist of variable names and/or numbers. Either a comma or a space can be used to separate the variables from each other.

When using probit or logistic regression, the dependent variable is always coded as 0 or 1. Regular multiple linear regression, with the dependent variable coded as 0 or 1, is inappropriate for the following reasons:

1. Estimated probabilities using multiple regression are not restricted to the interval (0,1). Using multiple linear regression, it is quite possible to get an estimated probability of greater than one or less than zero. It would be difficult to interpret this as an estimated probability. Unfortunately, it is quite common for 10% to 20% of the estimated probabilities to lie outside the unit interval when employing multiple regression with a (0,1) dependent variable.
2. Estimated probabilities using multiple regression are exceptionally sensitive to the observed distribution of the dependent variable (i.e., very small or very large mean for the dependent variable).
3. Standard multiple regression assumes that the effect of the independent variables is constant over the entire range of the predicted dependent variable. Probit and logistic regression, on the other hand, assume that the effects of the independent variables vary (i.e., nonlinear multiple regression).

In summary, there are two assumptions we make when using probit or logistic regression analysis:

1. the dependent variable of a record is assumed to be most flexible when its estimated probability is near one-half (i.e., the effect of an independent variable is expected to be highest when its estimate of probability is one-half).

In cases when the outcome of the event seems certain (e.g.  $P < .1$  or  $P > .9$ ) the explanatory variables have a smaller impact on changing the probability than the cases where the outcome is less certain. If the probability of an event is .9, we are in a stage of diminishing return to increasing its probability.

2. the effect of an independent variable depends on the estimated probability.

### ***Descriptive Statistics***

Descriptive statistics will be printed as part of the probit or logistic regression output if the option DS=Y is specified.

### Example of Descriptive Statistics Printout

Variables in the Analysis - Descriptive Statistics					
Variable		DV=0	DV=1	Overall	SD
DV1	Remission of Cancer				
V2	Cell Count	0.847	0.950	0.881	0.187
V3	Lithium	0.828	1.356	1.004	0.468
V4	Temperature	0.999	0.994	0.997	0.015

Number of valid cases = 27  
Number of cases where DV is 1 = 9  
Number of cases where DV is 0 = 18  
Number of missing cases = 0  
Response percent = 100.0 %  
Mean of dependent variable = 0.333

The example is an analysis looking at cancer remission data. The objective of the analysis is to assess the probability of complete cancer remission (REMISSION) on 3 patient characteristics (CELL, LITHIUM and TEMPERATURE).

The output reveals that there were 9 cases with cancer remission (DV=1) and 18 cases without cancer remission (DV=0), for a total of 27 cases. From the descriptive statistics, we can see that REMISSION appears associated with high mean values for CELL and LITHIUM, and a low mean value for TEMPERATURE.

### Simple Correlation Matrix

The simple correlation matrix can be requested with the option SC=Y. The simple correlation output can be used to examine the relationships between the independent variables.

### Example of a Simple Correlation Matrix Printout

Simple Correlation Matrix				
		DV1	V2	V3
V2	r	0.265		
	p	0.182		
V3	r	0.542	0.190	
	p	0.003	0.342	
V4	r	-0.156	0.108	-0.055
	p	0.436	0.591	0.786

### Regression Information

The regression coefficients and their standard errors are automatically included in each analysis. The output also includes the t statistic and its probability. The t statistic for an independent variable is its coefficient divided by its standard error.

### Example of a Regression Information Printout

<u>Probit Regression Coefficients</u>				
<u>Var.</u>	<u>Coeff.</u>	<u>Std. Error</u>	<u>T-Ratio</u>	<u>Prob.</u>
Const.	36.754	33.024	1.113	0.266
V2	5.629	4.418	1.274	0.203
V3	2.251	0.992	2.269	0.023
V4	-45.180	35.681	-1.266	0.205
Log of likelihood function = -10.95018				
Chi-square statistic for significance of equation = 12.471				
Degrees of freedom = 3				
Significance level = 0.006				

The chi-square statistic is used to measure the overall significance of the equation in explaining the dependent variable. This statistic is equivalent to the overall F-ratio in multiple regression and tests whether the set of independent variables as a group contributes significantly to the explanation of the dependent variable.

Estimates of the regression coefficients are obtained by maximizing the log of the likelihood function using the iterative Newton-Raphson method of scoring. Convergence is said to have occurred if the change in the log of the likelihood function on successive iterations is less than or equal to the tolerance level set in the parameter file. The tolerance level can be set between .001 and .000000001.

### Change in Probability Table

The change in probability table may be selected with the option PT=Y. As indicated earlier, the independent variable has its maximum effect when the estimated probability is one-half. Using the change in probability table, we can study how the probability changes when there is a change in the value of an independent variable.

### Example of a Change in Probability Table

<u>Change in Probability Table</u>						
<u>Var.</u>	<u>P=0.333</u>	<u>P=.1</u>	<u>P=.3</u>	<u>P=.5</u>	<u>P=.7</u>	<u>P=.9</u>
V2	2.047	0.988	1.957	2.246	1.957	0.988
V3	0.819	0.395	0.783	0.898	0.783	0.395
V4	-16.431	-7.928	-15.709	-18.024	-15.709	-7.928

The table reveals how a one unit increase in each independent variable will affect the probability of the dependent variable.

In the sample printout, note that if the estimated probability of cancer remission for an individual is .5, a one unit increase in the independent variable V3 is expected to increase the predicted probability by .89812. For that same individual, an increase of .1 in V3 would be expected to increase the predicted value of REMISSION by .89812 times .1 = .089812.

If the estimated probability of cancer remission for an individual is .9, a .1 increase in V3 is expected to increase the predicted probability by only .039501 (.39501 times .1). The first column in the "Change in Probability" table is always the effect of the independent variable evaluated at the sample mean of the dependent variable (.3333 for this example).

### ***Classification Table***

The classification table may be selected with the option CT=Y. It gives the frequency distribution of the observed value of the dependent variable (0 or 1) versus its predicted value based on the set of independent variables. If the dependent variable is well explained by the set of independent variables, we expect:

1. The frequencies in the first row of the table (observed value of DV=0) to be clustered in the first few columns.
2. The frequencies in the last row of the table (observed value of DV=1) to be clustered in the last few columns.

### ***Example of a Classification Table Printout***

<u>Classification Table</u>										
	Predicted									
Observed	0-.09	.1-.19	.2-.29	.3-.39	.4-.49	.5-.59	.6-.69	.7-.79	.8-.89	9-1.0
DV=0	9	2	4	0	0	1	1	0	1	0
DV=1	0	0	1	1	1	1	2	2	0	1
% Correct	100.0	100.0	80.0	0.0	0.0	50.0	66.7	100.0	0.0	100.0
Percent correct when DV is 0 = 83.3 % (n=18)										
Percent correct when DV is 1 = 66.7 % (n=9)										
Percent correct overall = 77.8 % (n=27)										
Chance based on frequency distribution = 66.7 %										
Percent improvement over chance = 11.1 %										

### ***Mean Substitution***

Mean substitution is one method often used to reduce the problem of missing information. Often, regression analysis is difficult because if one independent variable is not known, it is necessary to exclude the whole record from the analysis. It is possible that this can substantially reduce the number of records that are included in the analysis. Mean substitution overcomes this problem by replacing any missing independent variable with the mean of that variable. While this technique has the possibility of slightly distorting the results, it can make it possible to perform a regression with substantial missing data.

### ***Convergence Tolerance***

The convergence tolerance is used to find the maximum log of the likelihood function. It may be set using the option TL=n, where n is the convergence tolerance. A good initial value to use is .0000001.

The value of the convergence tolerance is very important. Too high a value does not result in the maximum of the likelihood function, while too small a value results in an iterative procedure which drifts about the maximum. If this is the case, the

program will start halving, working towards the previous (higher) value of the log of the likelihood function. A message will be printed as follows:

*Results are not based on last iteration but on a previous iteration which had a higher value for the log likelihood function. Convergence assumed after x iterations*

The above message will also be printed if the iterations don't converge. This is usually due to one (or more) of the following reasons:

1. Some of the explanatory variables are highly correlated. Examination of the correlation matrix in conjunction with collinearity diagnostics (such as those found in principal components and multicollinearity analyses) will usually indicate a variable which should have been omitted or transformed.
2. The response surface is very flat; this is usually due to the variables as a group being poor predictors of the dependent variable. There is then no significant maximum to find.
3. A variable may have very little variability and, therefore, be highly correlated with the constant term.
4. The iteration may go too far and skip the maximum point. This is usually due to setting the value too low. Although the message is printed, it is not usually a problem since StatPac always saves the value of the regression coefficients at the maximum value of the log likelihood and attempts halving towards the maximum point.

### ***Iteration Limit***

The maximum number of iterations can be set from 1 to 100 as a safeguard against a flat surface where iterations might proceed indefinitely. With the convergence tolerance set at 0.0000001, the number of iterations required for convergence usually varies from 4 to 12. Setting the maximum number of iterations around 30 is adequate. It should be noted that the data is read from disk with each iteration. The amount of time to execute one iteration is about the same as the amount of time it takes to run a multiple regression with the same number of variables and cases.

### ***Save Probabilities***

Researchers often want to save the probabilities in a file for further study. When SR=Y, you will be offered the opportunity to merge the predicted probabilities into the original data file at the completion of the analysis.

### ***Predict Interactively***

After performing a regression, you may want to predict values for the dependent variable. Select interactive prediction by entering the option PR=Y. At the completion of the analysis, you will then be prompted to enter a value for each independent variable, and the computer will use the regression coefficients and cumulative normal distribution (for probit) or the cumulative logistic distribution (for logit) to predict the probability that the value of the dependent variable is equal to one.

## Labeling and Spacing Options

Option	Code	Function
Labeling	LB	Sets the labeling for descriptive statistics to print the variable label (LB=E), the variable name (LB=N), or the variable number (LB=C).
Column Width	CW	Sets the minimum width of the columns (in inches).
Column Spacing	CS	Sets the spacing (in inches) between the columns.
Decimal Places	DP	Sets the number of decimal digits that will be shown.

---

## PCA Command

Principal components analysis (PCA) investigates relationships among variables without designating some as independent and others as dependent; instead PCA examines relationships within a single set of variables. The technique of PCA is primarily used to reduce the number of dimensions. Usually, most of the variation in a large group of variables can be captured with only a few principal components. Up to 200 variables can be analyzed.

This technique basically attempts to explain the variance-covariance structure of variables by constructing a smaller set of orthogonal (independent) linear combinations (principal components) of the original variables. The first principal component (PC) is that weighted combination of response variables which accounts for the maximum amount of total variation in the original variables. The second PC is that weighted combination of response variables which, among all combinations orthogonal to the first, accounts for the maximum amount of remaining variation.

The syntax to run a principal components analysis is:

**PCA <Variable list>**

For example, after conducting a lengthy survey, we might believe that several of the questions were actually measuring the same thing. Principal components could be used to isolate those questions that were measuring the same dimension. Take the following questions from a larger twenty-five question survey:

11. What is your annual income?
12. What percent of your salary do you pay in taxes?
13. How much discretionary income do you have?
14. What is the market value of your house?
15. How much disability insurance do you carry?

All of the above questions might be measuring a dimension related to income. If a principal components analysis extracted these variables into one component, we



might try to shorten future surveys by asking fewer questions about income. The command to run the principal components analysis would be:

PCA V1-V25

PC V1-V25      (Note: PCA may be abbreviated as PC)

Notice that all the questions in the survey were specified as part of the variable list. It is the job of PCA to extract the individual components.

A component is the weighted combination of variables which explains the maximum amount of remaining variation in the original variables (orthogonal to the previous components). That is, each component is mutually independent from all other components. Mathematically, the problem is one of explaining the variance-covariance structure of the variables through linear combinations of the variables. Primary interest lies in the algebraic sign and magnitude of the principal component coefficients (loadings), and in the total variation in the dependency structure explained by a component.

Grouping of variables is based on the magnitude of the loadings associated with each principal component. Loadings below .30 are usually disregarded for purposes of interpretation. Loadings are comparable to standardized partial regression coefficients. The sign and magnitude of each loading reveals how the particular variable is associated with that principal component. A loading may be interpreted like a correlation coefficient in that it shows the strength and direction of the relationship between a given variable and the principal component.

Generally, during study design, concepts or constructs are identified as part of the study goals. Variables are developed to answer the study goals. Principal components analysis can be used to evaluate how well each variable is associated with the construct it was designed to measure. In a well structured survey, each variable will have a high loading on only one construct (the one it was designed to measure). When a variable has a high loading on more than one principal component, the variable did not do a good job of discriminating between two or more constructs

### ***Principal Components***

Either the correlation matrix or covariance matrix may be used for deriving principal components. If the responses are in similar units, the covariance matrix has a greater statistical appeal. When the responses are in widely different units (age in years, weight in kilograms, height in centimeters, etc.) the correlation matrix should be used. In practice, the use of the correlation matrix is more common.

The PC option may be set to one of four values:

A = Correlation matrix

B = Covariance matrix with mean correction

C = Covariance matrix without mean correction

D = PCA not requested

### Example of Principal Components Printout

<u>Variance and Proportions</u>						
	Eigenvalue	Difference	Proportion	Cumulative		
PRIN1	4.603	3.428	76.72	76.72		
PRIN2	1.175	0.972	19.59	96.31		
PRIN3	0.203	0.188	3.39	99.70		
PRIN4	0.015	0.012	0.25	99.95		
PRIN5	0.003	0.002	0.04	99.99		
PRIN6	0.000	0.000	0.01	100.00		
Total variance = 6						
<u>Principal Component Loadings</u>						
	PRIN 1	PRIN 2	PRIN3	PRIN 4	PRIN 5	PRIN 6
V2	0.465	-0.001	-0.128	-0.052	-0.750	-0.450
V3	0.462	0.046	-0.196	-0.590	0.549	-0.312
V4	0.202	-0.798	0.562	-0.077	0.024	0.018
V5	0.321	0.596	0.728	0.008	0.009	0.107
V6	0.462	-0.053	-0.278	-0.122	-0.150	0.818
V7	0.462	-0.058	-0.149	0.793	0.338	-0.135

In the example above, PCA was run on variables 2 to 7 of the Longley data. The correlation matrix was used (PC=A) because the variables were in widely different units.

Note that the first two principal components account for 96.3% of the total variation in the dependency structure of the six variables. The first PC has equally high loadings on all variables except variable 3 (size of the armed forces). This component can be interpreted as an economic performance indicator. It is common for the first PC to load equally on most variables. The second component has high loadings on variable 3 (-.80) and variable 4 (.60) and the signs are different, implying that there is a high relationship between size of the armed forces and unemployment. The fact that the signs are different implies that as size of the armed forces goes up, unemployment goes down (as expected). The remaining four PC's account for less than 4% of the total variability and are ignored.

Following are some examples of the common uses of principal components analysis:

1. The most common use of principal components analysis is to derive a small number of linear combinations (principal components) from a set of variables that retain as much of the information in the original variables as possible. Often a small number of principal components can be used in place of the original variables for plotting, regression, clustering, etc.
2. Principal components analysis can also be viewed as an attempt to uncover approximate linear dependencies among variables (i.e., to understand the correlation structure). The multicollinearity diagnostics described below are based on the principal components.
3. It is often impossible to measure certain theoretical concepts, but many (highly interrelated) variables may be available to provide a mathematical formula for the theoretical concept. Principal components can be used to determine appropriate weights associated with each of these variables to provide an "optimum" measure of

a theoretical concept, such as mathematical ability. In essence, PCA obtains components which may be given special meaning.

4. Because the principal components are independent of each other, they may help to circumvent the problem of multicollinearity. Most of the variation in the variables is accounted for by the first few principal components. The last few principal components define dimensions of the regressor space that are not very prominent - the components are so flimsy that they can be blown around wildly by small perturbations in the data. To get rid of the instability of the estimates, you throw away the last few components. There are as many principal components as original variables, but usually only a small number of components are retained.

5. Principal components analysis is similar to factor analysis in that they both provide analysis of the interdependence structure of a set of variables. In factor analysis, it is assumed that each original variable is influenced by various factors. Some are shared by other variables in the set (common factors), while others are not shared by any other variable (unique factors). In PCA, on the other hand, no assumptions about the underlying structure of the variables are made. We define new hypothetical variables that are exact mathematical transformations of the original variables, but that are independent of each other. That is, we seek that set of linear combinations of the original variables that absorb and account for the maximum possible proportion of total variation in those variables.

The first principal component is the single best summary of the total variance; the second principal component is the best summary of the variance remaining after the first principal component has been extracted. Subsequent components are defined similarly until all the variance in the data is exhausted.

### ***Descriptive Statistics***

Descriptive statistics may be printed or suppressed using the DS=Y or DS=N option respectively.

### ***Example of Descriptive Statistics Printout***

<u>Variables in the Analysis - Descriptive Statistics</u>			
Variable		Mean	SD
V2	Year	1954.500	4.761
V3	Total Population	117.424	6.956
V4	Size of the Armed Forces	260.669	69.592
V5	Unemployment	319.331	93.446
V6	Gross National Product	387.698	99.395
V7	Gross Nat'l Prod Deflator	101.681	10.792
Number of Valid Cases = 16			
Number of Missing Cases = 0			
Response Percent = 100.0 %			

### ***Simple Correlation Matrix***

The simple correlation matrix (SC=Y) is the easiest way of examining linear dependencies between the variables. High intercorrelations between variables is a warning sign that collinearity might exist.

### Example of a Simple Correlation Printout

<u>Simple Correlation Matrix</u>			V2	V3	V4	V5	V6
V3	r		0.994				
	p		0.000				
V4	r		0.417	0.364			
	p		0.108	0.165			
V5	r		0.668	0.687	-0.177		
	p		0.005	0.003	0.511		
V6	r		0.995	0.991	0.446	0.604	
	p		0.000	0.000	0.083	0.013	
V7	r		0.991	0.979	0.465	0.621	0.992
	p		0.000	0.000	0.070	0.010	0.000

### Collinearity Diagnostics

Multicollinearity refers to the presence of highly intercorrelated predictor variables in regression models, and its effect is to invalidate some of the basic assumptions underlying their mathematical estimation. It is not surprising that it is considered to be one of the most severe problem in multiple regression models and is often referred to by social modelers as the "familiar curse". Collinearity diagnostics measure how much regressors are related to other regressors and how this affects the stability and variance of the regression estimates.

Signs of multicollinearity in a regression analysis include:

1. Large standard errors on the regression coefficient, so that estimates of the true model parameters become unstable and low t-values prevail.
2. The parameter estimates vary considerably from sample to sample.
3. Often there will be drastic changes in the regression estimates after only minor data revision.
4. Conflicting conclusions will be reached from the usual tests of significance (such as the wrong sign for a parameter).
5. Extreme correlations between pairs of variables.
6. Omitting a variable from the equation results in smaller regression standard errors.
7. A good fit not providing good forecasts.

We use the multicollinearity diagnostics:

1. To produce a set of condition indices that signal the presence of one or more near dependencies among the variables. (Linear dependency, an extreme form of multicollinearity, occurs when there is an exact linear relationship among the variables.)

2. To uncover those variables that are involved in particular near dependencies and to assess the degree to which the estimated regression coefficients are being degraded by the presence of the near dependencies.

In practice, if one independent variable has a high squared multiple correlation ( $r$ -squared) with the other independent variables, it is extremely unlikely that the independent variable in question contributes significantly to the prediction equation. When the  $r$ -squared is too high, the variable is, in essence, redundant.

When the collinearity analysis is requested with the CD=Y option, the statistics attributed to Belsley, Kuh and Welsch (1980) are printed (namely the eigenvalues, condition indices and the decomposition of the variances of the estimates with respect to each eigenvalue).

### Example of Collinearity Diagnostics

Collinearity Diagnostics								
	Eigenvalue	Condition Index	Portion V2	Portion V3	Portion V4	Portion V5	Portion V6	Portion V7
PRIN1	4.603	1.000	0.000	0.000	0.002	0.001	0.000	0.000
PRIN2	1.175	1.979	0.000	0.000	0.151	0.009	0.000	0.000
PRIN3	0.203	4.757	0.000	0.000	0.432	0.078	0.000	0.001
PRIN4	0.015	17.560	0.000	0.038	0.111	0.000	0.001	0.311
PRIN5	0.003	42.471	0.290	0.295	0.064	0.001	0.005	0.330
PRIN6	0.000	110.544	0.710	0.646	0.239	0.912	0.994	0.358
VIF			758.981	399.151	3.589	33.619	999.999	135.532

Note that variables 2,3,6 and 7 are highly correlated and the VIF's for all variables (except variable 4) are greater than 10 with one of them being greater than 1000.

Examination of the condition index column reveals a dominating dependency situation with high numbers for several indices. Further regressions on subsets of the independent variables are called for.

The following steps are generally recommended in diagnosing multicollinearity:

1. Inspection of the correlation matrix for high pairwise correlations; this is not sufficient, however, since multicollinearity can exist with no pairwise correlations being high.
2. VIF's greater than 10 are a sign of multicollinearity. The higher the value of VIF's, the more severe the problem. In the StatPac output, any VIF greater than 999.99999 is set to the value 999.99999.
3. Condition indices of 30 to 100 (generally indicating moderate to strong collinearities) combined with at least 2 high numbers (say greater than 0.5) in a "variance proportion" row are a sign of multicollinearity. The higher the condition indices, the more severe the multicollinearity problem. Three cases can be distinguished:

#### Case 1: Only one near dependency present

This occurs when only one condition index is greater than 30. A variable is involved in, and its estimated coefficient degraded by, the single near dependency if it is one of two or more variables in a row with "variance proportion" numbers in excess of some threshold value, such as .5 .

### Case 2: Competing dependencies

This occurs with more than one condition index of roughly the same magnitude and greater than 30. Here, involvement is determined by aggregating the "variance proportion" numbers of each variable over the high condition index rows. The variables whose aggregate proportions exceed 0.5 are involved in at least one of the dependencies, and therefore, have degraded coefficient estimates. The number of near dependencies present corresponds to the number of competing indices.

### Case 3: Dominating dependencies

Dominating dependencies exist when high condition indices (over 30) coexist with even larger condition indices. The dominating dependency can become the prime determinant of the variance of a given coefficient and obscure information about the simultaneous involvement in a weaker dependency. In this case, other variables can have their joint involvement obscured by the dominating near dependency. With this dominating near dependency removed, the obscured relationship may reappear. In this case, additional analysis, such as auxiliary regressions, is warranted to investigate the descriptive relations among all of the variables potentially involved.

Since the variance of any regression coefficient depends on regression residual error, sample size, and the extent of multicollinearity, the following are suggested as possibilities for increasing the precision of the regression coefficients:

1. Can the precision of measurement of any variable be improved? This has the effect of reducing regression residual error.
2. Can the model specification be improved? Have we, for example, omitted an important variable or transformed the variables appropriately (using logs, reciprocal, etc.) to match theory?
3. Can we increase the sample size, thereby decreasing mean square residual error?
4. Can we replace a variable with another less correlated with the current set of independent variables, but as correlated with the dependent variable?
5. A group of variables (highly intercorrelated) may be aggregated (or averaged), using principal components or factor analysis to find appropriate weights. This is especially true of variables measured in the same units, such as income.
6. Because multicollinearity indicates that some independent variables convey little information over that of the other variables, one way to scale down variables is to drop a redundant variable.

### ***Include Intercept***

A collinearity with the constant term occurs because some linear combination of two or more variables is essentially constant. This situation can be detected by omitting the intercept from the collinearity analysis (set with the IC=N option), and then examining the standard deviations or coefficients of variation (standard error/mean) of the variables.

### ***Variance Inflation Factors***

Variance inflation factors show the degree to which a regression coefficient will be affected because of the variable's redundancy with other independent variables. As the squared multiple correlation of any predictor variable with the other predictors approaches unity, the corresponding VIF becomes infinite.

For any predictor orthogonal (independent) to all other predictors, the variance inflation factor is 1.0. VIF<sub>i</sub> thus provides us with a measure of how many times

larger the variance of the *i*th regression coefficient will be for multicollinear data than for orthogonal data (where each VIF is 1.0). If the VIF's are not unusually larger than 1.0, multicollinearity is not a problem. An advantage of knowing the VIF for each variable is that it gives a tangible idea of how much of the variances of the estimated coefficients are degraded by the multicollinearity. VIF's may be printed using the VI=Y option.

### **Save Scores**

Researchers often want to create principal component scores for respondents. This provides an indication of how strongly an individual loads on each component. The data is first standardized and the principal component loadings are used as coefficients for calculating the component score.

You can save the component scores with the options command SS=Y. At the completion of the analysis you will be given the opportunity to merge to component scores with the original data file..

### **Mean Substitution**

Mean substitution is one method often used to reduce the problem of missing information. Usually, if one independent variable is not known, it is necessary to exclude the whole record from the analysis. It is possible that this can substantially reduce the number of records that are included in the analysis. Mean substitution overcomes this problem by replacing any missing variable with the mean of that variable. While this technique has the possibility of slightly distorting the results, it can make it possible to perform a principal components analysis with substantial missing data.

### **Labeling and Spacing Options**

Option	Code	Function
Labeling	LB	Sets the labeling for descriptive statistics to print the variable label (LB=E), the variable name (LB=N), or the variable number (LB=C).
Column Width	CW	Sets the minimum width of the columns (in inches).
Column Spacing	CS	Sets the spacing (in inches) between the columns.
Decimal Places	DP	Sets the number of decimal digits that will be shown.

---

## **FACTOR Command**

Factor analysis is similar to principal components analysis. It is another way of examining the relationships between variables. Factor analysis differs from principal components in that there are usually fewer factors than variables. Up to 200 variables may be included.

StatPac contains two different methods for extracting factors from a set of variables: varimax and powered-vector. Both methods extract factors that are independent (orthogonal) from other factors. This is known as a simple structure analysis. The program also contains an option to improve the simple structure by allowing the

factors to be correlated with each other. When factors are correlated, it is known as an oblique reference structure analysis.

The syntax to run a factor analysis is:

**FACTOR** <Variable list>

**FA** <Variable list> (Note: **FACTOR** may be abbreviated as **FA**)

Factor analysis is essentially a way of examining the correlation matrix. While both techniques (varimax and powered-vector) are quite different from each other in methodology, they have the same first step, which is to decide what values are to be used for the communality estimates. For each variable, the communality is defined as the sum of the squares of all the factor loadings for that variable. A factor loading itself can be thought of as the correlation of a variable with the factor. The communalities are placed on the diagonal of the correlation matrix.

### ***Initial Communality Estimates for the Diagonal***

There are three commonly used communalities: units, absolute row maximums and squared multiple correlation coefficients. Units is probably the most commonly used since it is always a value of one. That is, the value of one is placed on the diagonal of the correlation matrix. The highest absolute row maximums refers to the highest absolute correlation coefficient in each row. The squared multiple correlation coefficients refers to the correlation of each variable with the remainder of the variables. That is, the diagonal variable is regressed against all other variables, and the coefficient of multiple correlation is placed on the diagonal.

You can select the initial diagonal values by setting the DI option to one of three values:

1 = Use units, (i.e., 1)

2 = Use highest absolute row correlation

3 = Use squared multiple correlation coefficient

### ***Type of Solution***

There are two different techniques to extract factors. The varimax solution (TY=V) is most commonly used. The first step with the varimax technique is known as "principal factor analysis". It is the same as a principal components analysis except that the extraction of principals is stopped by some predetermined criterion. When a sufficient number of principals has been extracted, a rotational technique called "varimax" is used to create the simple structure factor loadings.

The powered-vector technique (TY=P) uses an entirely different approach. This approach is faster than the varimax technique because it does not require a principal components analysis first. Instead, it uses a cluster technique to extract factors directly from the correlation matrix. An additional technique called "weighted cross-factor" rotation is often used with the powered-vector solution to provide a cleaner separation among the factors.



### Example of a Factor Analysis Printout

<u>Orthogonal Power Vector Simple Structure Factor Loadings</u>					
		FACT1	FACT2		
V2		0.999	0.017		
V3		0.996	-0.037		
V4		0.403	0.910		
V5		0.683	-0.550		
V6		0.994	0.060		
V7		0.992	0.075		

<u>Variance and Proportions</u>					
	<u>Communalities</u>		<u>Variance</u>	<u>Proportion</u>	<u>Cumulative</u>
V2	0.998	FACTOR1	4.591	76.5	76.5
V3	0.993	FACTOR2	1.141	19.0	95.5
V4	0.990				
V5	0.769				
V6	0.992				
V7	0.989				

### Descriptive Statistics

Occasionally, you may be interested in the means and variance for each of the variables in the analysis. Descriptive statistics may be printed or excluded from the output by using the options DS=Y and DS=N, respectively.

### Example of a Descriptive Statistics Printout

<u>Variables in the Analysis - Descriptive Statistics</u>			
Variable		Mean	SD
V2	Year	1954.500	4.761
V3	Total Population	117.424	6.956
V4	Size of the Armed Forces	260.669	69.592
V5	Unemployment	319.331	93.446
V6	Gross National Product	387.698	99.395
V7	Gross Nat'l Prod Deflator	101.681	10.792

Number of Valid Cases = 16			
Number of Missing Cases = 0			
Response Percent = 100.0 %			

### Simple Correlation Matrix

It is often desirable to print the simple correlation matrix when performing a factor analysis (SC=Y). It can provide a good initial understanding of the interrelationships in the data.

### ***Example of a Simple Correlation Matrix Printout***

<u>Simple Correlation Matrix</u>		V2	V3	V4	V5	V6
V3	r	0.994				
	p	0.000				
V4	r	0.417	0.364			
	p	0.108	0.165			
V5	r	0.668	0.687	-0.177		
	p	0.005	0.003	0.511		
V6	r	0.995	0.991	0.446	0.604	
	p	0.000	0.000	0.083	0.013	
V7	r	0.991	0.979	0.465	0.621	0.992
	p	0.000	0.000	0.070	0.010	0.000

### ***Principal Components Analysis***

The results of the principal components analysis may be printed (PC=Y) or not (PC=N). This option only refers to printing the output. The remaining principal components options must be set appropriately regardless of the PC setting.

### ***Cross-Factor Rotation***

Cross-factor rotation is often used in conjunction with the powered-vector technique to improve upon the simple structure. It usually provides a "cleaner" structure, that is, a clearer separation of the factors. Select cross-factor rotation with the option CR=Y.

### ***Oblique Simple Structure Factor Loadings***

After performing either a varimax or powered-vector solution, you may want to perform an oblique rotation. Both the varimax and powered-vector solutions make an arbitrary assumption that factors are unique and independent of one another. The oblique rotation removes this restriction and allows factors to be correlated with each other. The oblique simple structure factor loadings may be printed (OS=Y) or excluded (OS=N).

The oblique rotation is often used to test the uniqueness of factors. If the resulting factors have low intercorrelations after an oblique rotation, it is fairly certain that the factors are orthogonal (independent of each other).

### ***Example of Oblique Simple Structure Factor Loadings***

<u>Oblique Simple Structure Factor Loadings</u>		
	FACT1	FACT2
V2	0.904	0.103
V3	0.922	0.047
V4	0.000	0.946
V5	0.866	-0.447
V6	0.879	0.139
V7	0.873	0.158

### ***Oblique Factor Correlation Matrix***

The correlation matrix of the factors can be printed (OC=Y) or excluded (OC=N). It represents relationships between factors after the oblique rotation. If the correlations are low (less than .3), we can be confident that the varimax or powered-vector solution produced unique and unrelated factors.

### ***Example of an Oblique Correlation Matrix Printout***

<u>Oblique Factor Correlation Matrix</u>	
	FACT1
FACT2	-0.849

### ***Number of Factors***

When the EX=1, StatPac will extract NF factors from the data. This is often used to test a specific hypothesis that has been developed. For example, you might believe that twenty items on a survey are really measuring only two major factors. You could test this hypothesis by using the option:

**OPTIONS EX=1 NF=2**

In this case, only two components would be extracted. They could then be examined as to how well they "hang together". The only time you use the NF option is to test a specific hypothesis.

### ***Percent of Total Variance to Explain***

When the EX=2, StatPac will continue to extract factors until it has accounted for PR proportion of the variance.

For example, to use the powered-vector technique and extract factors until 95 percent of the variance has been accounted for, enter the option:

**OPTIONS EX=2 PR=95**

### ***Minimum Variance Proportion for Principal Inclusion***

When EX=3, StatPac will continue to extract components until the next component would not account for a minimum proportion of the total variance.

For example, if you set MP=5, the program will extract all the components that account for at least five percent of the total variance. Any component that does not account for five percent of the variance will not be included in the analysis.

### ***Varimax Rotational Factor Angle***

The varimax rotational technique is one of the better factoring methods. It usually provides a clearer separation of factors than other methods. The technique extracts factors using normalized factor loadings during the iterations. These factors are assumed to be unique (orthogonal) and not correlated with each other. The technique attempts to maximize the variance of the squared loadings for each factor. A factor loading itself can be thought of as the correlation of a variable with the factor.

Using this method, the angle of rotation is calculated in each iteration. When the angle is less than the varimax rotational factor angle (RF), the process is completed. In other words, each iteration is a rotation in an attempt to improve the simple structure. When the rotational angle is less than the value of RF, the exit criteria has been achieved. The most often used value is one degree. The RF option can be used to set the angle to any value. For example, the following options command sets the exit criteria to one and a half degrees:

**OPTIONS RF=1.5**

### ***Convergence Tolerance for Principal Components***

The convergence tolerance is the way that the "resolution" of the components is controlled. In other words, it determines the point at which the program decides it has finished extracting a component. Setting the convergence tolerance too low will result in a very large number of calculations and, in the worst case, may cause the program to exceed the limit on the number of iterations. Setting the value too high would cause the program to prematurely believe it had extracted a component. A good starting value for the convergence tolerance is TL=.000001.

### ***Convergence Tolerance for Oblique Rotation***

Each iteration in the oblique rotation improves the simple structure by successively decreasing amounts. The convergence tolerance (OT) places a limit on the convergence process, and is used as the exit criteria for oblique rotation. The iterations will continue until additional rotations do not improve the simple structure by more than the convergence tolerance value. While the convergence tolerance could be any number greater than zero, a typical value might be .000001. The options command to set this value is:

**OPTIONS OT=.000001**

### ***Iteration Limit***

The maximum number of iterations for any of the factoring algorithms is included to limit the time that the computer could be working on the problem. Convergence usually occurs in fewer than ten iterations; however, as the convergence tolerance is set to lower values, it will require more iterations to achieve a solution. Generally, the maximum number of iterations is set to 100 (IT=100). This seems to allow most solutions, and at the same time, prevents unreasonable calculation times.

### ***Save Scores***

Researchers often want to create factor scores for respondents. This provides an indication of how strongly an individual loads on each factor. The data is first standardized and the factor loadings are used as coefficients for calculating the factor scores. It should be noted that the loadings will be those of last rotation requested with the other options. Thus, if an oblique rotation was requested (i.e., the last rotation to be performed) the saved scores will be those obtained from the oblique rotation.

You can save the factor scores with the options command SS=Y. At the completion of the analysis, you will be given the opportunity to merge the saved scores with the original data file.

### ***Mean Substitution***

Missing data can become a problem when there are few cases. In the worst case, missing data may make it impossible to perform an analysis. Mean substitution is one method often used to combat the problem of missing data. When mean substitution is used (MS=Y), any data that is missing is replaced with the mean of the variable.

### ***Labeling and Spacing Options***

Option	Code	Function
Labeling	LB	Sets the labeling for descriptive statistics to print the variable label (LB=E), the variable name (LB=N), or the variable number (LB=C).
Column Width	CW	Sets the minimum width of the columns (in inches).
Column Spacing	CS	Sets the spacing (in inches) between the columns.
Decimal Places	DP	Sets the number of decimal digits that will be shown.

---

## **CLUSTER Command**

The objective of cluster analysis is to separate the observations into different groups (clusters) so that the members of any one group differ from one another as little as possible, whereas observations across clusters tend to be dissimilar. The grouping can be used to summarize the data or as a basis for further analysis. In discriminant analysis, the groups are already defined, whereas in cluster analysis the purpose is to define the groups.

The syntax of the command to run cluster analysis is:

**CLUSTER <Variable list>**

As an example, a researcher studying iris flowers wanted to know if the iris would group into types based on length and width of their sepals and petals. The four clustering variables are:

V1	SEPAL LENGTH	length of sepal
V2	SEPAL WIDTH	width of sepal
V3	PETAL LENGTH	length of petal
V4	PETAL WIDTH	width of petal

The command to run cluster analysis could be specified in several ways:

**CLUSTER SEPAL LENGTH, SEPAL WIDTH,  
PETAL LENGTH, PETAL WIDTH**  
**CLUSTER V1 V2 V3 V4**  
**CLUSTER V1-V4**  
**CL V1-V4** (Note: **CLUSTER** can be abbreviated as **CL**)

In the first example, a continuation line was used to extend the variable list. The variable list can consist of variable labels and/or variable numbers. Either a comma or a space can be used to separate the variables from each other.

**CLUSTER** provides two types of cluster analysis: agglomerative hierarchical cluster analysis and non-hierarchical cluster analysis. For hierarchical methods, the general procedure is as follows:

1. Begin with as many clusters as there are observations (i.e., each cluster consists of exactly one observation).
2. Search for the most similar pair of clusters. This involves evaluating a criterion (distance) function for each possible pair of clusters and choosing the pair of clusters for which the value of the criterion function is the smallest. The criterion function is constructed using the clustering variables; the actual formula for the criterion function depends on the clustering algorithm used. Label the chosen clusters as p and q.
3. Reduce the number of clusters by one through the merger of clusters p and q; the new cluster is labeled q.
4. Perform steps 2 and 3 until all the observations are in one cluster. At each stage, the identity of the merged clusters as well as the value of the criterion function is stored.

Non-hierarchical clustering methods begin with the number of clusters given. Their primary use is to refine the clusters obtained by hierarchical methods. The non-hierarchical cluster analysis methods used by StatPac are convergent K-means methods and generally follow the following sequence of steps:

1. Begin with an initial partition of data units into clusters. There are several different initial partitions available in StatPac.

2. Take each data unit in sequence and compute the distances to all cluster centroids; if the nearest centroid is not that of the data unit's current cluster, then reassign the data unit and update the centroids of the losing and gaining clusters.
3. Repeat step 2 until convergence is achieved; that is, continue until a full cycle through the data set fails to cause any changes in cluster membership.

Cluster analysis performed on small data sets (a few hundred cases) will run relatively fast. However, the time to run the analysis increases exponentially with the number of cases in the data file. When thousands of cases are involved, it may take several hours to complete the analysis. Therefore, during preliminary analyses on large data sets, it may be desirable to use a SELECT statement to limit the number records being analyzed.

### ***Type Of Clustering Algorithm***

There are six different clustering algorithms available in StatPac. The TY option is used to select the clustering method. Algorithms 1-3 are agglomerative hierarchical clustering algorithms while algorithms 4-6 are non-hierarchical clustering algorithms. Each of the six algorithms are described below:

#### Minimum average sum of squares cluster analysis TY=1

With this algorithm, the clusters merged at each stage are chosen so as to minimize the average contribution to the error sum of squares for each member in the cluster. This quantity is also the variance in each cluster and is similar to average linkage in that it tends to produce clusters of approximately equal variance. Consequently, if the clusters are all of approximately the same density, then there will be a tendency for large natural groups to appear as several smaller clusters, or for small natural groups to merge into larger clusters.

#### Ward's method TY=2

At each stage, this method minimizes the within-cluster sum of squares over all partitions due to the merger of clusters p and q. This method tends to join clusters with a small number of observations and is biased towards producing clusters with roughly the same number of observations.

#### Centroid method TY=3

This method minimizes the squared Euclidian distance between clusters at each stage. The centroid method is not as sensitive to the presence of outliers, but does not perform as well as the first two methods if there are no outliers.

If there are no outliers, one of the first two methods should be used. The first method performs better than Ward's method under certain types of errors (Milligan, 1980). The three non-hierarchical clustering algorithms are all based on the convergent K-means method (Anderberg, 1973) and differ only in terms of their starting values.

#### Convergent K-means using minimum average sum of squares centroids TY=4

This algorithm first runs the minimum average sum of squares hierarchical cluster analysis method and uses the centroids from this method as input to the convergent K-means procedure. The distance measure used to allocate an observation to a cluster in the convergent K-means procedure is the Euclidian distance obtained from the clustering variables for that observation.

#### Convergent K-means using Ward method centroids TY=5

This algorithm first runs the Ward hierarchical cluster analysis method and uses the centroids from this method as input to the convergent K-means procedure. The distance measure used to allocate an observation to a cluster in the convergent K-

means procedure is the Euclidian distance obtained from the clustering variables for that observation.

#### Convergent K-means using centroids from the centroid method TY=6

This algorithm first runs the centroid hierarchical cluster analysis method and uses the centroids from this method as input to the convergent K-means procedure. The distance measure used to allocate an observation to a cluster in the convergent K-means procedure is the Euclidian distance obtained from the clustering variables for that observation.

Non-hierarchical methods generally perform better than hierarchical methods if non-random starting clusters are used. When random starting clusters are used (for example, the first p observations are used as centroids for the p starting clusters), the non-hierarchical clustering methods perform rather poorly. The random start methods were, therefore, not implemented in StatPac. K-means procedures appear more robust than any hierarchical methods with respect to the presence of outliers, error perturbations of distance measures and choice of distance metric. However, non-hierarchical methods require the number of clusters to be given. Many studies recommend the following series of steps in running cluster analysis:

1. Run cluster analysis using one of the first two hierarchical cluster analysis algorithms (minimum average sum of squares or Ward methods).
2. Remove outliers from the data set. Outliers can be located by looking at the distance from the cluster centroids (CC option), or the hierarchical tree diagram (one observation clusters that are late in merging with other clusters). Outliers often represent segments of the population that are under-represented and therefore, should not be discarded, without examination.
3. Delete dormant clustering variables. These can be located using the decomposition of sum of squares (DC option).
4. Determine the number of clusters. This can be done using the criterion function column in the decomposition of sum of squares (DC option), as well as the hierarchical tree diagram (TD option).
5. Once outliers are discarded, dormant variables omitted and the number of clusters determined, run one of the first two non-hierarchical methods (TY=4 or 5) several times, varying the number of clusters.

### ***Number of Clusters***

In the first cluster analysis run on a data set, you should choose one of the first three hierarchical clustering algorithms, and set the number of clusters equal to 99 (NC=99). This will print a hierarchical tree diagram and the decomposition of sum of squares, leaving the other options off (i.e., TD=Y, DC=Y, CC=N, CM=N). Note that the type of clustering option (TY) should be 1, 2 or 3 (one of the hierarchical clustering algorithms).

Once you have examined the hierarchical tree diagram and the decomposition of sum of squares, you would select the number of clusters using the NC option. Cluster analysis is an exploratory technique, and you will probably have to rerun the cluster analysis several times, varying the number of clusters as well as the clustering algorithm and the set of clustering variables. It is not uncommon to set the number of clusters to a few more than you suspect there are clusters, in an attempt to discover outliers. Non-hierarchical clustering algorithms are more effective in spotting outliers by this method than their hierarchical counterparts.



## ***Descriptive Statistics***

Descriptive statistics may be printed or suppressed using the DS=Y or DS=N option, respectively.

### ***Example of Descriptive Statistics Printout***

<u>Variables in the Analysis - Descriptive Statistics</u>			
<u>Variable</u>		<u>Mean</u>	<u>SD</u>
V1	Preference Rating	20.773	45.933
V2	Reluctance Rating	21.727	72.148
Number of Valid Cases = 22			
Number of Missing Cases = 0			
Response Percent = 100.0 %			

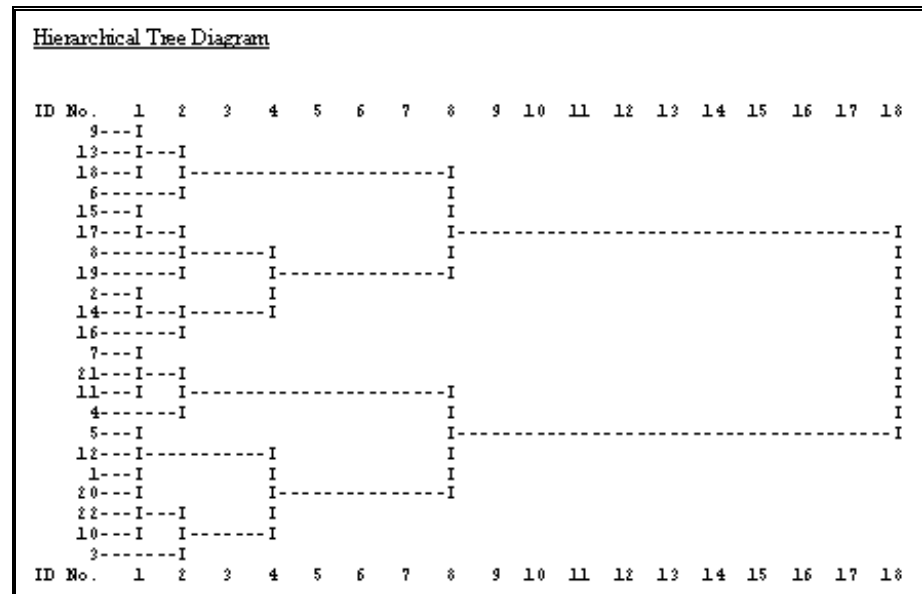
## ***Hierarchical Tree Diagram***

The hierarchical tree diagram provides the analyst with an effective visual condensation of the clustering results. The hierarchical tree diagram is one of the most commonly used methods of determining the number of clusters. It is also useful in spotting outliers, as these will appear as one-member clusters that are joined later in the clustering process.

The numbers at the top and bottom of the hierarchical tree diagram represent equally-spaced values of the criterion function. It gives a pictorial representation of the criterion function information.

If two or more clusters in a set of data are distinguished very well from each other, all merges but the last few (where "true" clusters are joined) will be clumped to the left of the tree diagram because of the extreme dissimilarity of the "true" clusters (i.e., most of the criterion function is accounted for by these clusters). To better understand the internal structure of these "true" clusters, it may be necessary to rerun cluster analysis separately on each of these "true" clusters.

## Example of a Hierarchical Tree Diagram



## Cluster Centroids

The cluster centroids are simply the means of each clustering variable for each cluster. The cluster centroids are probably the most useful multivariate characterization of the clusters.

## Example of a Cluster Centroids Printout

<u>Cluster Size Table</u>		
Cluster #	Size	
1	4	
2	7	
3	3	
4	8	

<u>Cluster Centroids</u>		
Cluster	V1	V2
1	-18.250	-42.250
2	69.286	-38.286
3	40.333	141.333
4	-9.500	61.375

## Decomposition of Sum of Squares

This option combines two types of information: criterion function information and decomposition of sum of squares information.

The criterion function is useful in determining the number of clusters. It is expressed, at each clustering stage, as a proportion of the value of the criterion function when all observations are joined in one cluster (the last stage in a hierarchical cluster analysis). The criterion function, at a given clustering stage, is a measure of the distance between all observations in all clusters. Consequently, at the start (when there are as many clusters as observations), the value of the criterion function is zero (each cluster has zero variance because it contains only 1 observation). As clusters are joined, the value of the criterion function increases. The criterion function rises slowly in the first stages, as the most similar clusters are joined and there is very little within cluster variability. However, as true distinct groups are joined, the within-cluster variability increases and the criterion function rises sharply.

As an example, suppose you are analyzing a data set with four clearly defined groups (clusters). The value of the criterion function should rise very slowly until you reach three clusters, in which case two "true" clusters are joined. This would be the clue as to the number of clusters (i.e., the sharp rise in the criterion function when you reach three clusters).

A "random" variable (variable not useful in separating clusters) can have a detrimental effect in cluster analysis and should be eliminated. One way of evaluating the relationship between a given hierarchical classification, and each of the clustering variables, is through the examination of the growth in unexplained sum of squares, as the clustering progresses through increasing levels of aggregation.

At the beginning of clustering, each observation is represented perfectly by the mean vector to which it belongs and there is no within cluster error. At the highest level of aggregation, there is only one cluster and it contains every observation. The proportion of unexplained sum of squares is, therefore, 1.0. At any stage between these two extremes, within cluster error sum of squares is that portion of the total variance unexplained by the current set of clusters.

To locate "random" variables, one compares the step by step growth in the proportion of unexplained sum of squares for each clustering variable. For a few variables, the fractions may remain small up to the last few stages, whereas, for other variables, the fractions may get large at a fairly early stage. The former variables may be thought of as being dominant in the results, while the latter are dormant. Repeating the clustering with dormant variables eliminated should have little effect on the results. However, deleting a dominant variable probably will have a marked influence on the clustering.

This kind of analysis can be an especially useful device for generating a parsimonious set of variables, to be used in subsequent attempts to cluster the data. It should be noted that if the data set has very distinct clusters, the unexplained sum of squares will rise slowly, even when a variable is dominant. It is not until the size of the clusters increases and/or "true" clusters are joined that the proportion of unexplained sum of squares rises sharply.

Indirectly, the decomposition of sum of squares can also be used as an indicator of the number of true clusters. As this option generates a line for each observation, the number of clustering variables "decomposed" is restricted to what will fit on one line. This option is only possible with hierarchical clustering algorithms (TY=1, 2, or 3). The decomposition of sum of squares will remain the same, regardless of the number of clusters chosen, if you use the same data set and the same clustering algorithm.

### ***Example of A Decomposition of Sum of Squares Printout***

<u>Criterion Function and Decomposition of Sum of Squares</u>			
Clusters	Crit. Fnc.	V1	V2
21	0.001	0.000	0.001
20	0.002	0.000	0.003
19	0.003	0.003	0.003
18	0.004	0.006	0.004
17	0.006	0.010	0.004
16	0.007	0.011	0.006
15	0.010	0.017	0.007
14	0.013	0.023	0.009
13	0.018	0.041	0.009
12	0.024	0.048	0.014
11	0.031	0.048	0.024
10	0.039	0.053	0.034
9	0.048	0.082	0.034
8	0.058	0.082	0.048
7	0.070	0.123	0.049
6	0.085	0.161	0.054
5	0.113	0.173	0.089
4	0.166	0.299	0.112
3	0.292	0.422	0.240
2	0.419	0.862	0.240
1	1.000	1.000	1.000

### ***Cluster Membership and Distance to Cluster Centroids***

This option will list the members of each cluster as well as the Euclidian distance of each member from its cluster centroid. This output provides useful information on how homogeneous the clusters are, and provides an aid in the detection of outliers.

### ***An Example of a Cluster Membership and Distance to Centroids Printout***

<u>Breakdown of Cluster Membership and Euclidian Distance to Centroid</u>		
<u>Cluster</u>	<u>Case #</u>	<u>Distance</u>
1	9	12.314
1	13	22.553
1	18	26.002
1	6	33.958
2	15	16.023
2	17	13.980
2	8	44.536
2	19	46.441
2	2	30.778
2	14	35.721
2	16	48.095
3	7	13.864
3	21	9.758
3	11	14.067
4	5	28.109
4	12	38.832
4	1	58.053
4	3	58.231
4	10	34.618
4	20	13.524
4	22	13.588
4	4	52.620

### ***Standardize Clustering Variables***

Standardizing the clustering variables consists of subtracting the variable mean, and dividing by the variable standard deviation. The data should be standardized if the clustering variables are in widely different units (age in years, weight in kilograms, height in centimeters, etc.) to avoid giving variables with higher variances more weight in the clustering process. With the hierarchical clustering algorithms especially, standardizing the clustering variables tends to reduce the effect of outliers on the final clustering solution.

### ***Mean Substitution***

Mean substitution is one method often used to reduce the problem of missing information. Often, cluster analysis is difficult because if one clustering variable is not known, it is necessary to exclude the whole record from the analysis. It is possible that this can substantially reduce the number of records that are included in the analysis. Mean substitution overcomes this problem by replacing any missing clustering variable with the mean of that variable. While this technique has the possibility of slightly distorting the results, it can make it possible to perform a cluster analysis with substantial missing data. If an observation for which one or more missing clustering variables was replaced by the mean shows up as an outlier, then this observation should be eliminated from future cluster analysis runs.

### ***Iteration Limit***

The non-hierarchical methods require that the maximum number of iterations be specified. The non-hierarchical clustering process will stop either when the maximum number of iterations has been reached, or when an iteration has resulted in no observation being moved from one cluster to another. Since the non-hierarchical clustering algorithms in StatPac start with hierarchical cluster centroids, they will rarely require more than a few iterations. Setting the maximum number of iterations to ten should be sufficient in most cases. Should the iteration limit be reached, the iteration summary output should provide guidance as to how much the iteration limit should be increased.

### ***Save Cluster Membership Variable***

Cluster analysis is usually a first step for running other statistical techniques. The next step in the analysis often involves one of the following:

1. Run analysis within each of the clusters.
2. Run discriminant analysis on the clusters, thus allowing one to get multivariate statistics on the clusters as well as a plot of the first two canonical axis.

The SM option allows you to save the cluster membership variable for further analysis of the clusters. At the end of the analysis you will be given the opportunity to merge the cluster membership variable into the original data.

### ***Labeling and Spacing Options***

Option	Code	Function
Labeling	LB	Sets the labeling for descriptive statistics to print the variable label (LB=E), the variable name (LB=N), or the variable number (LB=C).
Column Width	CW	Sets the minimum width of the columns (in inches).
Column Spacing	CS	Sets the spacing (in inches) between the columns.
Decimal Places	DP	Sets the number of decimal digits that will be shown.

---

## **DISCRIMINANT Command**

Discriminant function analysis is a technique for the multivariate study of group differences. It is similar to multiple regression in that both involve a set of independent variables and a dependent variable. In multiple regression, the dependent variable is a continuous variable, whereas in discriminant analysis, the dependent variable (often called the grouping variable) is categorical.

Discriminant analysis can be seen as an extension of probit or logistic regression. In probit and logistic regression, the dependent variable is numerically coded as 0 or 1; in discriminant analysis the grouping variable may be numeric or alpha (e.g., 1, 2, 3 or A, B, C). When there are only two groups, many researchers use probit or logistic regression and code the two groups as 0 and 1.

Discriminant analysis is used to:

1. describe, summarize and understand the differences between groups.

2. determine which set of independent variables best captures or characterizes group differences.
3. classify new subjects into groups or categories.

Canonical correlation analysis (an option) can be used to reduce the dimensionality of the independent variables, similar to principal components analysis. Canonical analysis also makes it possible to determine how well the groups are separated, using two linear combinations of the independent variables in the discriminant equations.

The syntax of the command to run a stepwise discriminant analysis is:

**DISCRIMINANT** <Dependent variable> <Independent variable list>

The maximum number of categories (groups) for the dependent variable is 24. Up to 200 independent variables may be specified.

As an example, a researcher studying three types of iris flowers wanted to know if the type of iris could be determined from the length and width of their sepals and petals. The IRIS TYPE variable (V1) is coded 1=Setosa, 2=Versicol and 3=Virginic. Note that this is a categorical variable; an iris is one type or another (we normally won't have a mixed breed iris).

We also have four independent variables: SEPAL LENGTH (V2) is the length of sepal, SEPAL WIDTH (V3) is the width of sepal, PETAL LENGTH (V4) is the length of petal, and PETAL WIDTH (V5) is the width of petal.

The command to run the discriminant analysis could be specified in several different ways:

**DISCRIMINANT IRIS TYPE, SEPAL LENGTH, SEPAL WIDTH,  
PETAL LENGTH, PETAL WIDTH**

**DISCRIMINANT V1 V2 V3 V4 V5**

**DISCRIMINANT IRIS TYPE V2-V5**

**DI V1-V5** (Note: DISCRIMINANT can be abbreviated as DI)

In each example, the dependent variable (IRIS TYPE) was specified first, followed by the independent variable list. The variable list itself can consist of variable labels and/or variable numbers. Either a comma or a space can be used to separate the variables from each other.

The dependent variable may be alpha or numeric. If it is numeric, it must be coded 1 through 24. If it is alpha, it must be coded A, B, C, etc. If the study design does not contain value codes and labels for the dependent variable, the program will use the data itself to determine the value codes.

A discriminant function equation is used to obtain the posterior probability that an observation came from each of the groups. An observation is, therefore, classified, by the discriminant analysis, into the group with the highest posterior probability as estimated above.

### ***Descriptive Statistics***

The mean and standard deviations for all the independent variables can be printed with the descriptive statistics option (DS=Y). The output contains the means and

standard deviations for each of the independent variables. When DS=C, the output will contain descriptive statistic controlled for each of the dependent variable groups.

### ***Example of a Descriptive Statistics Printout***

<u>Descriptive Statistics for All Groups</u>			
Variable		Mean	SD
V2	Sepal Length	58.500	8.253
V3	Sepal Width	30.533	4.364
V4	Petal Length	37.827	17.580
V5	Petal Width	12.087	7.585
Number of Valid Cases = 150			
Number of Missing Cases = 0			
Response Percent = 100.0 %			

### ***Simple Correlation Matrix***

The within group correlation matrix is obtained by pooling the correlation matrix from each of the groups (SC=Y). If two variables are highly correlated, it is possible that the matrices are not well conditioned, and it might be beneficial to run the discriminant analysis again without one of the variables. If Wilks' lambda does not show a significant increase, you might want to leave the variable out of the discriminant analysis.

### ***Example of a Within Group Correlation Matrix Printout***

<u>Within Groups Correlation Matrix</u>				
		V2	V3	V4
V2	r	0.530		
	p	0.000		
V3	r	0.748	0.370	
	p	0.000	0.000	
V4	r	0.363	0.466	0.493
	p	0.000	0.000	0.000

### ***Group Discriminant Function Coefficients***

The group discriminant function (classification) coefficients can be printed with the CO option (CO=Y). The output includes a constant and a coefficient for each independent variable, for each value of the grouping variable. Each coefficient provides an estimate of the effect of that variable (in the units of the raw score) for classifying an observation into each group.



### **Example of Group Discriminant Function Coefficients Printout**

<u>Group Classification Function Coefficients</u>			
	1	2	3
V2	2.286	1.582	1.303
V3	2.345	0.713	0.388
V4	-1.532	0.500	1.179
V5	-1.699	0.635	2.071
Const.	-85.258	-72.827	-103.471

### **Classification Matrix**

The classification matrix may be selected with the option CM=Y. It gives the frequency distribution of the observed group versus its predicted group, based on the set of independent variables in the discriminant function. This option also calculates the percent correctly classified in each group, as well as over all groups. If the group is well predicted by the set of independent variables, we expect to find most observations falling on the diagonal of this matrix (i.e., observations in group i would be classified as belonging to group i). The classification matrix also provides valuable insight into which groups are well separated and which groups are harder to separate.

### **Example of a Classification Matrix**

<u>Classification Matrix (Number of cases classified in each group)</u>				
Group	% Correct	1	2	3
1	100.0	49	0	0
2	94.1	0	48	3
3	98.0	0	1	49
Total	97.3	49	49	52

### **List Incorrectly Classified Cases**

This listing may be selected with the option IC=Y. For each case that was incorrectly classified, this option gives the case number, the group that case came from as well as the predicted group based on the independent variables in the discriminant function. This listing can be used to check for errors in one or more of the independent variables.

### Example of Incorrectly Classified Cases Listing

#### Incorrect Classifications

Record 1 from group 2 was incorrectly classified in group 3  
Record 5 from group 3 was incorrectly classified in group 2  
Record 9 from group 2 was incorrectly classified in group 3  
Record 12 from group 2 was incorrectly classified in group 3

### Print Each Step

You can print the statistics for each step using the option PS=Y. This may be important when you want to study how the inclusion or deletion of a variable affects other variables.

Wilks' lambda is the ratio of the determinants of the within groups cross-product to total cross-product matrices. It has values between 0 and 1. Wilks' lambda is similar to the coefficient of multiple determination (r-squared) in multiple regression, except that it moves in the opposite direction. Where r-squared gets larger as the equation improves, Wilks' lambda gets smaller as the equation improves. Thus, Wilks' lambda could be interpreted as the proportion of variance in the dependent variable that is not explained by the discriminant function model. Large values of Wilks' lambda indicate that the independent variables in the equation are not doing a good job of predicting the dependent variable group. Small values of Wilks' lambda indicate good separation between (at least some) groups.

The overall F-ratio measures whether the variables in the equation are useful in classifying cases. Typically, a probability of .05 or less leads us to reject the hypothesis that the discriminant function does not improve our ability to classify cases. The F-to-enter value for any variable not in the equation tests whether adding this variable in the equation would lead to a significant decrease in Wilks' lambda, while the F-to-remove value for any variable in the equation tests whether this variable would lead to a significant increase in Wilks' lambda. These values are used to determine the independent variable to enter or delete in the next step.

### Example of the Print Each Step Output

<u>Step1</u>	Variable V4 Entered	Wilks Lambda =0.061	F(2,147)=1139.125, p=0.000
	<u>Var.</u>	<u>F-to-Remove</u>	<u>F-to-Enter</u>
	V2		32.025
	V3		42.583
	V4	1139.125	
	V5		25.032
<u>Step2</u>	Variable V3 Entered	Wilks Lambda =0.038	F(4,292)=300.102, p=0.000
	<u>Var.</u>	<u>F-to-Remove</u>	<u>F-to-Enter</u>
	V2		10.841
	V3	42.583	
	V4	1066.185	
	V5		34.366

### Summary Table

A good way to get an overview of how the steps proceeded, and what effect each step had upon Wilks' lambda, is to print the summary table. To print the summary table, use the option ST=Y.

### Example of a Summary Table

<u>Stepwise Discriminant Function Analysis Summary Table</u>					
Step No.	Variable		Wilks Lambda	F-Value to Enter/Remove	No. of IVs
	Entered	Removed			
1	V4		0.061	1139.125	1
2	V3		0.038	42.583	2
3	V5		0.026	34.366	3
4	V2		0.025	3.919	4

### Canonical Variable Analysis

Canonical analysis can be used to reduce the dimensionality of the independent variables, and is similar to principal components.

The first canonical variable is the linear combination of independent variables that best summarizes the differences among the groups. The second canonical variable is the next best linear combination orthogonal to the first one, and so on. You can print the canonical variable analysis by entering the option CV=Y. This option provides two tables.

The first table is a summary of the eigenvalues associated with each canonical variable, as well as the proportion of the "between-group variability" accounted for by each canonical variable.

### Example of a Canonical Variable Summary Table

<u>Canonical Variable Summary Table</u>				
	Eigenvalue	Proportion	Cumulative	Canonical Correlation
CANO1	30.437	99.05	99.05	0.984
CANO2	0.291	0.95	100.00	0.475
Total variance = 30.729				

The second table gives the coefficients of the canonical variables. This is similar to the eigenvalue print-out in principal components. The number of canonical variables reported is the lesser of (the number of groups minus 1) and the number of variables entered in the discriminant function.

### **Example of a Canonical Variable Coefficients Table**

<u>Canonical Variable Coefficients</u>		
Variable	CANO1	CANO2
V2	-0.075	0.007
V3	-0.155	0.212
V4	0.209	-0.097
V5	0.281	0.289

### **Canonical Variables Evaluated at Group Means**

You can select to print a table of the canonical variables, evaluated at the group means using the option GM=Y.

### **Example of Canonical Variables Evaluated at Group Means Printout**

<u>Canonical Variables Evaluated at Group Means</u>		
Group	CANO1	CANO2
1	-7.512	0.220
2	1.754	-0.725
3	5.572	0.523

### **Save Canonical Pair**

Researchers often want to save the first two canonical variables for future analysis. You can save them with the options command SP=Y. At the completion of the analysis, you will be given the opportunity to merge the canonical variable pair and predicted group into the original data.

### **Prior Probabilities**

Prior probabilities are the initial values that will be placed on the diagonal of the matrix. Prior probabilities may be set to equal (PP=E), automatic (PP=A), or individual probabilities may be specified.

When PP=E, the values for the prior probabilities will be equal to one divided by the number of categories (so each category has an equal prior probability).

Setting the prior probabilities option to automatic (PP=A) will assign prior probabilities to each category, based on the frequency of that category. The sum of the prior probabilities will be one.

The other method of setting the prior probabilities is to explicitly specify them. When this method is used, a prior probability must be assigned to each category. They do not need to sum to one. The following option would assign prior probabilities to three alpha categories. Note that commas are used to separate them from each other.

OPTIONS PP=(A=3.5, B=4.7, C=2.9)

### ***Number of Variables to Force***

The ability to force variables into an equation is important for several reasons:

1. A researcher often wishes to replicate the analysis of another study and, therefore, to force certain core variables into the equation, letting stepwise discriminant analysis choose from the remaining set.
2. Some variables may be cheaper or easier to measure, and the user may want to see whether the remaining variables add anything to the equation.
3. When independent variables are highly correlated, one of them may be more accurate than the rest, and you may want to force this variable in the equation.

The syntax for the number of variables to force in is FO=n, where n is the number of variables to force. For example, FO=3 will force the first 3 variables from the independent variable list into the equation. An option such as FO=200 may be used to perform a non-stepwise analysis (all variables will be included in the analysis).

### ***Category Creation***

The actual categories (dependent variable groups) can be created either from the study design value labels (CC=L) or from the data itself (CC=D). When the categories are created from the labels, the value labels themselves will be used to define the dependent variable groups. Any data that does not match up with a value label (e.g., miskeyed data) will be counted as missing. When categories are created from the data, all data will be considered valid, whether or not there is a value label for it.

### ***F to Enter & F to Remove***

When faced with a large number of possible explanatory variables, two opposed criteria of selecting variables for a discriminant analysis are usually involved:

1. To make the equation useful for classification purposes, we would like our model to include as many of the independent variables as possible so that reliable group classification can be determined.
2. Because of the costs involved in obtaining information on a large number of independent variables, we would like the equation to include as few of the independent variables as possible.

The compromise between these two extremes is generally called "selecting the best set of independent variables". This involves multiple execution of discriminant analysis, in an attempt to add variables to improve classification or remove variables to simplify the classification equations. Stepwise discriminant analysis provides a partial automation of this procedure.

An important property of the stepwise procedure is based on the fact that a variable may be indicated to be significant in an early stage and, thus, be entered in the equation. After several other variables are added to the equation, however, the initial variable may be indicated to be insignificant (redundant), and thus removed from the model. This method is often referred to as forward inclusion with backward elimination.

The algorithm used by StatPac is as follows:

1. First, enter into the discriminant analysis all variables which the user wishes to force into the equation.
2. Enter the predictor that produces the greatest decrease in Wilks' lambda from all the remaining predictors whose entry is not inhibited by the F-to-enter.
3. Remove the predictor that makes the least increase in Wilks' lambda from all (non-forced) predictors whose removal is not inhibited by the F-to-remove.

Note that step 2 is executed only when it is not possible to execute step 3. If neither can be executed, the stepping is complete.

The following should be considered when setting F-to-enter and F-to-remove values in the parameter table:

1. A variable is removed if the F-value associated with that variable is less than the F-to-remove value set in the parameter table. Similarly, a variable is added if the F-value associated with that variable would be greater than the F-to-enter value set in the parameter table, if that variable were entered in the current equation.
2. Care should be taken to ensure that the F-to-remove be less than the F-to-enter; otherwise, a variable would be entered and then removed at alternate steps.
3. The default values for the F-to-enter and F-to-remove for many mainframe packages, and StatPac, are 4.0 and 3.9, respectively, which provide useful starting values.
4. Setting the F-to-remove value low yields the forward inclusion method.
5. For the first run on a data set, it is common to set the F-to-enter and F-to-remove values low, to execute a large number of steps.

### ***Steps Limit***

The steps limit is the maximum number of steps that can occur. Each inclusion and deletion of a variable is counted as one step. The purpose of the steps limit option is to limit computer time. The syntax for the steps limit option is SL=n, where n is the maximum number of steps allowed.

### ***Mean Substitution***

Mean substitution is one method often used to reduce the problem of missing information. Often, discriminant analysis is difficult because if one independent variable is not known, it is necessary to exclude the whole record from the analysis. It is possible that this can substantially reduce the number of records that are included in the analysis. Mean substitution overcomes this problem by replacing any missing independent variable with the mean of that variable. While this technique has the possibility of slightly distorting the results, it can make it possible to perform a discriminant analysis with substantial missing data. If the value of the dependent (grouping) variable is missing, the whole record is deleted from the analysis.

### ***Predict Interactively***

After performing a discriminant analysis, you may want the posterior probabilities associated with each group, for a new observation, or an observation in your data for which the value of the dependent variable was missing. This is known as interactive prediction.

You can select interactive prediction by entering the option PR=Y. At the end of the analysis you will then be prompted to enter a value for each independent variable, and the computer will use the discriminant function coefficients to estimate the posterior probabilities associated with each group. The observation is then assigned

to the group with the highest posterior probability. You can skip over any independent variable by just pressing <enter>; the value of the group mean for that independent variable will be used.

---

## ANOVA Command

The analysis of variance procedure provides a systematic way of studying variability. Usually, we are interested in how much of the variability of scores on the dependent variable can be explained by the differences between scores (levels) on the experimental variables (factors). StatPac may contain up to three factors and up to 90 levels for each factor.

### ***Type of Design***

StatPac Gold contains eleven different ANOVA designs (or models). Choosing the appropriate design for a particular experiment requires careful evaluation. It is quite possible to perform an inappropriate statistical procedure by choosing the wrong model. Since StatPac has no way of knowing that the model is wrong, it will produce erroneous results. The following types of models are available:

1. One Factor Completely Randomized Design
2. Randomized Complete Block Design
3. Randomized Complete Block Design With Sampling
4. Two Factor Factorial in Completely Randomized Design
5. Two Factor Factorial in Randomized Complete Block Design
6. Three Factor Factorial in Completely Randomized Design
7. Three Factor Nested Design
8. Split-Plot With Completely Randomized Design of Main Plots
9. Split-Plot With Randomized Complete Block Design of Main Plot
10. Split-Plot With Sub-Units Arranged in Strips
11. Latin Square Design

The number relating to the type of design has no intrinsic meaning in and of itself. It is simply a number used to specify which model you want StatPac to use for the analysis.

### ***Missing Data in ANOVA Designs***

Many analysis of variance experimental designs involve assigning an equal number of cases to each cell. When all cells do not contain the same number of cases, the model is said to be "unbalanced". Unbalanced designs usually occur because of differential attrition (e.g., some of the crop dies, respondents become unavailable or refuse to participate, recording errors, etc.).

StatPac uses an unweighted means solution when there is not an equal number of cases in each cell. This involves the use of the harmonic mean to adjust the sums of squares. If there is an equal number of observations in each cell, the unweighted means solution is equivalent to the usual least squares approach. The unweighted means solution requires at least one case in every cell. There may not be any cells where all the data is missing.

The unweighted means approach is an approximation technique and does not produce exact results when the design is unbalanced. Although the F statistics may not be exact, researchers have found that the F-ratios are acceptable unless the design is highly unbalanced. As a measure of departure from a balanced design, use the ratio of  $N_i/N_j$ , where  $N_i$  is the greatest number of observations in any cell and  $N_j$  is the minimum. A ratio of 4:1 is tolerable, but a ratio of 16:1 should not be accepted. In these cases, an exact solution can be obtained by creating appropriate dummy variables and performing a regression analysis. For a detailed discussion, see D.G. Gosslee and H.L. Lucas (Biometrics, Volume 21, p. 115-133).

## ***Command Syntax and the Data File Structure***

The syntax for the ANOVA command is similar for all eleven different models. The only difference is in the number of factors that are specified as part of the command. There are two general forms of the command:

ANOVA (Type) <Dependent Variable> (<Fact. 1>) (<Fact. 2>) (<Fact. 3>)

This format is used when each record contains a single value for the dependent variable. The type of design is specified first and must be enclosed in parentheses. It may be a number between one and eleven. The dependent variable is specified next. Finally, a variable is specified for each factor. Each of these variables is also enclosed in parentheses. If a design only contains one (or two) factors, only one (or two) need be specified.

The second form of the ANOVA command is used when each record contains several values for the dependent variable (one for each level of one of the factors). In this case, the dependent variable is not a single variable, but rather a variable list. Since any one of the factors may be the dependent variable list, the syntax may take on three different forms:

ANOVA (Type) (<Factor 1 var. list>) (<Factor 2>) (<Factor 3>)

ANOVA (Type) (<Factor 1>) (<Factor 2 var. list>) (<Factor 3>)

ANOVA (Type) (<Factor 1>) (<Factor 2>) (<Factor 3 var. list>)

Which form of the ANOVA command you use depends upon how the data file is arranged. The following three examples illustrate the different forms of the command.

The first example is a completely randomized one-way design (Type 1). There is only one factor for this kind of model. As stated above, there are two possible formats for the command. They are:

ANOVA (1) <Dependent variable> (<Factor 1>)

ANOVA (1) (<Factor 1 variable list>)

For example, let's say we are interested in studying the effect of training after one hour, two hours, and three hours. There are two ways the data file might be



organized. In the first format, each record contains the score and the time of measurement. It would appear like this:

```
34 1 (record 1 - score at time 1 for case 1)
43 2 (record 2 - score at time 2 for case 1)
55 3 (record 3 - score at time 3 for case 1)
41 1 (record 4 - score at time 1 for case 2)
52 2 (record 5 - score at time 2 for case 2)
63 3 (record 6 - score at time 3 for case 2)
37 1 (record 7 - score at time 1 for case 3)
59 2 (record 8 - score at time 2 for case 3)
61 3 (record 9 - score at time 3 for case 3)
```

In this example, the dependent variable (SCORE) is variable one and the factor (TIME-PERIOD) is variable two. The values for variable two represent the levels of the variable (i.e., what time the score was taken). Two examples of the syntax to run a one-way ANOVA using the above data file would be:

```
ANOVA (1) V1 (V2)
ANOVA (1) SCORE (TIME-PERIOD)
```

In the other type of data file format, each record contains the score for each hour of training. With this format, the above data file would appear like this:

```
34 43 55 (rec 1 - score after each hour of training for case 1)
41 52 63 (rec 2 - score after each hour of training for case 2)
37 59 61 (rec 3 - score after each hour of training for case 3)
```

This data file format differs from the previous format, but the data is the same. The dependent variable is no longer just held in a single variable. There is a dependent variable for each level of the factor. Variable one is the score for time one (TIME-1), variable two is the score for time two (TIME-2), and variable three is the score for time three (TIME-3).

Three examples of the syntax to run a one-way analysis of variance for this type of data file format are:

```
AN (1) (V1-V3) (Note: ANOVA may be abbreviated as AN)
ANOVA (1) (V1,V2,V3)
ANOVA (1) (TIME-1, TIME-2, TIME-3)
```

The only difference between the two data files is in the way in which they are coded. The actual data is the same in both files, and the results of the analysis of variance will be identical.

The second example is two-way ANOVA in a completely randomized design (Type 4). A two-way analysis of variance is used to examine the effect that two

independent variables have on the dependent variable. Because of the high cost of conducting experiments and the possibility of interaction effects, researchers often use a two-way design to get the most out of each experiment.

For example, let's say we are studying the GROWTH (dependent variable) of four different hybrid SEEDS. We are also interested in whether any brand of FERTILIZER works better than the others. Instead of conducting two separate experiments, we conduct only one and analyze the results with a two-way ANOVA. This has an added advantage because it will take into account the interaction between fertilizer and seed type.

The first form of the command syntax for the two-way ANOVA is:

ANOVA (4) <Dependent variable> (<Factor 1>) (<Factor 2>)

Since there are two factors in this design, the second form of the command could specify either factor as the variable list:

ANOVA (4) (<Factor 1 variable list>) (<Factor 2>)

ANOVA (4) (<Factor 1>) (<Factor 2 variable list>)

Notice that the last two forms of the command are variations of the same syntax. Since there are two factors, they both must be specified in the command syntax. The actual syntax depends upon the way the data file is structured.

The first form of the command is used when there is a dependent variable and a variable for each factor. A sample data file format for this experimental design might look like this:

```
67 1 A  (record 1 - yield this acre is 67
        fertilizer used is coded as 1
        type of hybrid seed is coded as A)
54 2 A  (record 2 - yield this acre is 54
        fertilizer used is coded as 2
        type of hybrid seed is coded as A)
3 A    (record 3 - yield this acre is missing - crop died
        fertilizer used is coded as 3
        type of hybrid seed is coded as A)
52 1 B  (record 4 - yield this acre is 52
        fertilizer used is coded as 1
        type of hybrid seed is coded as B)
61 2 B  (record 5 - yield this acre is 61
        fertilizer used is coded as 2
        type of hybrid seed is coded as B)
27 3 B  (record 6 - yield this acre is 27
        fertilizer used is coded as 3
        type of hybrid seed is coded as B)
```

Two commands to run a two-way ANOVA with this data file are:

```
ANOVA (4) V1 (V2) (V3)
ANOVA (4) GROWTH (FERTILIZER) (SEED)
```

The data file could contain the same information formatted in another way. For example, the following data file contains the same information as the previous file except that the dependent variable (GROWTH) is specified for each type of fertilizer.

```
67 54 A (record 1 - growth for all 3 fertilizers with seed type A)
52 61 27 B (record 2 - growth for all 3 fertilizers with seed type B)
```

Variable one is the growth for FERTILIZER-1, variable two is the growth for FERTILIZER-2, and variable three is the growth for FERTILIZER-3. Variable four is the SEED type. Three commands to perform a two-way ANOVA with this type of data file format are:

```
ANOVA (4) (V1, V2, V3) (V4)
ANOVA (4) (V1-V3) (V4)
ANOVA (4) (FERTILIZER-1, FERTILIZER-2, FERTILIZER-3) (SEED)
```

For a final example of data file formatting, we'll look at a typical three-factor factorial experiment (Type = 6). This model is similar to the previous model except that three experimental factors are being examined.

There are now four possible data file formats corresponding to four forms of syntax:

```
ANOVA (6) <Dep. var.> (<Factor A>) (<Factor B>) (<Factor C>)

ANOVA (6) (<Factor A variable list>) (<Factor B>) (<Factor C>)

ANOVA (6) (<Factor A>) (<Factor B variable list>) (<Factor C>)

ANOVA (6) (<Factor A>) (<Factor B>) (<Factor C variable list>)
```

A medical researcher was studying the pain-relieving properties of two different drugs (A and B). In addition to testing the drugs themselves, she wanted to compare high and low doses as well as the method of administration (oral or intravenously). The dependent variable is a measure of pain RELIEF (on a scale of 0 to 9), factor A is the DRUG type, factor B is the DOSE and factor C is the METHOD of administration.

In the first data file format, the dependent variable and each factor represent unique variables in the data file.

8 A H O (Pain relief 8 - drug A - high dose - oral admin.)  
 6 A L O (Pain relief 6 - drug A - low dose - oral admin.)  
 9 A H I (Pain relief 9 - drug A - high dose - iv. admin.)  
 8 A L I (Pain relief 8 - drug A - low dose - iv. admin.)  
 5 B H O (Pain relief 5 - drug B - high dose - oral admin.)  
 4 B L O (Pain relief 4 - drug B - low dose - oral admin.)  
 6 B H I (Pain relief 6 - drug B - high dose - iv. admin.)  
 5 B L I (Pain relief 5 - drug B - low dose - iv. admin.)

The commands to run the ANOVA with this data file organization are:

ANOVA (6) V1 (V2) (V3) (V4)  
 ANOVA (6) RELIEF (DRUG) (DOSE) (METHOD)

The last three forms of the command are used when one of the factors is specified as part of a variable list. For example, both methods of administering the drug could be contained in the same data record. The data file would appear like this:

A H 8 9 (Drug A - high dose - relief for oral & iv admin.)  
 A L 6 8 (Drug A - low dose - relief for oral & iv admin.)  
 B H 5 6 (Drug B - high dose - relief for oral & iv admin.)  
 B L 4 5 (Drug B - low dose - relief for oral & iv admin.)

The commands to run the ANOVA with this data file format would be:

ANOVA (6) (V1) (V2) (V3, V4)  
 ANOVA (6) (DRUG) (DOSE) (ORAL, IV)

It is especially important to match the appropriate command syntax with the format of the data file. The versatility of StatPac to read several different data file formats makes it easy to analyze most data sets. The best advice is to plan the analysis before entering data.

### ***Descriptive Statistics***

Descriptive statistics for each cell may be printed or suppressed with the option DS=Y or DS=N, respectively. In multi-factor experiments it is often desirable to print descriptive statistics for each factor controlled for the other factors. To print controlled descriptive statistics, use the option DS=C. The descriptive statistics printout will contain the count, mean and unbiased standard deviation.

## Example of Descriptive Statistics

<u>Design: Two factor factorial in completely randomized design</u>				
Factor A (Fixed): Source of Protein				
Level 1 -- 1=Beef				
Level 2 -- 2=Cereal				
Level 3 -- 3=Pork				
Factor B (Fixed): Level of Protein				
Level 1 -- 1=High				
Level 2 -- 2=Low				
Descriptive Statistics for: Weight Gain in Grams				
Factor A: Source of Protein				
Factor B: Level of Protein				
<u>Cell Definition</u>		<u>N</u>	<u>Mean</u>	<u>SD</u>
(A) Level 1	(B) Level 1	10	100.000	15.136
(A) Level 1	(B) Level 2	9	78.000	14.169
(A) Level 2	(B) Level 1	10	85.900	15.022
(A) Level 2	(B) Level 2	10	83.900	15.709
(A) Level 3	(B) Level 1	10	99.500	10.916
(A) Level 3	(B) Level 2	8	74.875	15.075

## Anova Table

The ANOVA table is the heart of the analysis. The F-test reveals whether or not there are significant differences between the levels of the experimental factor(s). The actual terms that appear in the ANOVA table depend on the type of design.

Generally, experiments involve assigning cases to groups on the basis of some experimental condition and observing the differences between the groups on the dependent variable. As the differences between the groups increase, so will the F-ratio. The actual formula for a particular F-test depends upon the ANOVA design and whether the factors are fixed or random.

A significant F-ratio means that there is a significant difference between the means of the dependent variable for at least two groups. For example, in a completely randomized two-factor factorial analysis (Type = 4), there are three F-ratios: one for the A factor, one for the B factor and one for the AB interaction. Their interpretation is as follows:

1. The F-ratio for factor A tests whether the factor A variable has a significant effect on the response of the dependent variable, averaged over all levels of the factor B variable. A significance level less than .05 is generally considered significant.
2. The F-ratio for factor B tests whether the factor B variable has a significant effect on the response of the dependent variable, averaged over all levels of the factor A variable.
3. The F-ratio for interaction tests whether there is significant interaction between the factor A and factor B variables. Interaction results from the failure of differences

between responses at the different levels of one of the variables to remain constant over the different levels of the other variable. If the interaction term is significant, the F-test for factor A and B should be interpreted with care. The next step is generally to examine the means for all pairs of levels of the two variables.

### Example of an Anova Table

<u>ANOVA Summary Table</u>					
Source of Variation	DF	Sum of Squares	Mean Squares	F-Ratio	Significance Level
A	2	159.267	79.633	0.384	0.683
B	1	3716.946	3716.946	17.912	0.000
AB (Interaction)	2	1444.372	722.186	3.480	0.038
Error	51	10583.175	207.513		
Total	56	15903.760			

### Fixed and Random Factors

When conducting tests of significance in multi-factor designs, you must specify whether each of the factors is fixed or random. This will determine which of the mean squares in the analysis of variance table is used for the denominator of the F-ratio.

The concept of whether a factor is fixed or random can be determined using the following reasoning. Assume a factor has a potential (or population) of P levels which may be quite large. The experimenter may group the P potential levels into p effective levels by either combining adjoining levels or deliberately selecting what are considered to be representative levels. While p is less than P, the effective levels still represent the entire potential (or population). Whenever the selection of the p levels from the potential P levels is determined by some systematic non-random procedure, the factor is fixed. In the special case where the number of levels (p) of a factor is equal to P (no levels were grouped), the factor is also fixed.

Examples of fixed factors include rates of application, varieties, types of compound, etc. With fixed factors, we are generally interested in estimating fixed effects associated with the specific levels of the fixed factors.

In contrast to this systematic selection procedure, if p levels of a factor included in the experiment represent a random sample from the potential P levels, the factor is considered to be a random factor. For example, if a random sample of p of the P potential hospitals is included in the experiment, the factor (hospitals) is a random factor. In most practical situations in which random factors are encountered, p is quite small relative to P.

Examples of random factors include people, herds, plants, lots, hospitals, etc. With random factors, we are generally interested in estimating the variability present in these factors.

The F1, F2 and F3 options may be used to set a factor as fixed or random. For example, to set factors 2 and 3 as random factors, you would use the option:

**OPTIONS F2=R F3=R**

The formulas for the F tests depend upon the type of design and whether the factors are fixed or random. If a factor is inappropriately specified as random, StatPac will simply continue processing it as if it were random and will be unable to detect the error.

### ***Critical F Probability***

The analysis of variance by itself can reveal that differences exist between different levels of the experimental condition. That is, a significant F-ratio indicates a significant difference between at least two of the levels. It does not actually tell where the differences occur. The lsd t-tests between all the combinations of means reveal where the actual difference(s) is (are). The t-tests will only be performed if the F-ratio is significant at the critical F probability. For example, if the critical F probability is equal to .05 (CF=.05), the t-tests will be performed only if the F-ratio is less than or equal to .05.

The t-tests will be performed for only those pairs of means that have a significant F-ratio. Take the example where factor A has a significant F-ratio and factor B does not. The lsd t-tests will be performed between all combinations of levels of factor A, while no t-tests will be performed between the levels of factor B.

### ***Critical T Probability***

If an F-ratio is significant at the critical F probability, StatPac will run through the pair of cell means and compute the lsd t-statistic and probability of t. We are usually only interested in those combinations where the t-statistic is significant. The critical t probability allows the selective printing of the t-statistics depending on the probability of t. For example, if the critical t probability is set to .05 (CT=.05), only those t-values that have probabilities of .05 or less will be printed.

The t statistic will reveal differences between the group means. If any t is significant, it will be printed. This procedure, called the new lsd (least significant difference) t-test, is considered to be one of the most conservative post-hoc tests.

### ***Example of a t-Test Printout***

Post-hoc lsd t-tests between group means - (Values of p are for a two-tailed test.)	
Note: Statistics are printed only if p is less than or equal to .05	
t(51)=4.154, p=0.000	Factor (B) Level 1 Factor (B) Level 2
t(51)=3.324, p=0.002	Factor (A) Level 1 & Factor (B) Level 1 Factor (A) Level 1 & Factor (B) Level 2
t(51)=2.189, p=0.033	Factor (A) Level 1 & Factor (B) Level 1 Factor (A) Level 2 & Factor (B) Level 1

### ***Category Creation***

The actual categories (levels for each factor) can be created either from the study design value labels (CC=L) or from the data itself (CC=D). When the categories are created from the labels, the value labels themselves will be used to define the levels for each factor. Any data that does not match up with a value label (e.g., misspelled

data) will be counted as missing. When categories are created from the data, the data itself will be used to define the levels, whether or not there is a matching value label.

### ***Print Codes***

The code for each level can be printed or suppressed with the PC=Y and PC=N options, respectively.

### ***Kruskal-Wallis Test***

The non-parametric equivalent of an analysis of variance is the Kruskal-Wallis test. The data is ranked, and the sum of the ranks for each of the groups is used to calculate the statistic. The probability is determined using the chi-square distribution with the degrees of freedom equal to the number of groups minus one.

### ***Labeling and Spacing Options***

Option	Code	Function
Labeling	LB	Sets the labeling for descriptive statistics to print the variable label (LB=E), the variable name (LB=N), or the variable number (LB=C).
Column Width	CW	Sets the minimum width of the columns (in inches).
Column Spacing	CS	Sets the spacing (in inches) between the columns.
Decimal Places	DP	Sets the number of decimal digits that will be shown.

### ***Anova Examples***

The following pages give a brief description of the eleven analysis of variance designs which StatPac can analyze along with simple examples and the statistical tests for each of these designs. It is important to note that, in many cases, more than one design may be appropriate for a given data set.

#### **1. One Factor Completely Randomized Design**

Syntax:

ANOVA (1) <Dependent variable> (<Factor A>)

ANOVA (1) (<Factor A variable list>)

Discussion: This is the simplest design and the easiest to carry out. The design contains only one factor, and can handle unequal numbers of observations per level.

An Example: In an attempt to study fat absorption in doughnuts, 24 doughnuts were prepared (six doughnuts from each of four kinds of fats). The dependent variable is grams of fat absorbed, and the factor variable is the type of fat. The factor contains four levels (four types of fat were tested). The researcher accidentally dropped one of the doughnuts from the second type of fat, so the second type of fat contains five observations instead of six.



The analysis of variance table follows:

<u>ANOVA Summary Table</u>					
Source of Variation	DF	Sum of Squares	Mean Squares	F-Ratio	Significance Level
A	3	1504.435	501.478	4.722	0.013
Error	19	2018.000	106.211		
Total	22	3522.435			

## **2. Randomized Complete Block Design**

Syntax:

ANOVA (2) <Dependent variable> (<Factor A>) (<Factor B>)

ANOVA (2) (<Factor A variable list>) (<Factor B>)

ANOVA (2) (<Factor A>) (<Factor B variable list>)

Note: Factor A is always the experimental treatment

Factor B is always the replication or block

Discussion: This design is easy to carry out. It is essentially a one-way analysis of variance with replications (blocks). This design always contains exactly one observation per cell. Units assigned to the same block are as similar as possible in responsiveness, thus increasing the precision of treatment comparisons by eliminating block-to-block variation. Blocks can represent time, location or experimental material. Examples of blocks include repeated testing over time, litter-mates, and groups of experimental plots as similar as possible in terms of fertility, drainage, and liability to attack by insects.

An Example: A researcher wants to study the effects four seed treatments and a control group (a total of five treatment levels) on the germination of soybean seeds. The factor variable is the type of treatment (1 to 5). Five germination beds were prepared for each level of treatment and 100 seeds factors were planted in each bed. Thus, the replications are the five beds. The dependent variable is the number of plants in each bed which failed to germinate. There are two factors: treatment and replication.

The analysis of variance table follows:

<u>ANOVA Summary Table</u>					
Source of Variation	DF	Sum of Squares	Mean Squares	F-Ratio	Significance Level
Replications	4	49.840	12.460		
A	4	83.840	20.960	3.874	0.022
Error	16	86.560	5.410		
Total	24	220.240			

### **3. Randomized Complete Block Design With Sampling**

Syntax:

```
ANOVA (3) <Dep. var.> (<Factor A>) (<Factor B>) (<Factor C>)  
ANOVA (3) (<Factor A variable list>) (<Factor B>) (<Factor C>)  
ANOVA (3) (<Factor A>) (<Factor B variable list>) (<Factor C>)  
ANOVA (3) (<Factor A>) (<Factor B>) (<Factor C variable list>)
```

Note:     Factor A is always the experimental treatment  
          Factor B is always the replicate or block  
          Factor C is always the sampling or determinations

Discussion: This design is the same as the previous design with more than one observation per experimental unit. Experiments often contain more than one observation per experimental unit when the researcher wishes to estimate the reliability of measurement. With this design, the error term is broken down into experimental error and sampling error. Sampling error measures the failure of observations made on any experimental unit to be precisely alike. Experimental error is usually expected to be larger than sampling error. In other words, variation among experimental units is expected to be larger than variation among subsamples of the same unit.

An Example: The objective of an experiment was to study the effect of corn variety on protein content. A field was divided into three similar plots. Each plot was subdivided into fourteen sections. Fourteen different varieties of corn were planted in each plot (one in each section). After harvest, two protein determinations were made on each variety of corn in each plot. The dependent variable was the protein content. Factor A was the type of corn and contained 14 levels. Factor B was the replication (the three plots). Factor C was the two determinations.

The analysis of variance table follows:

<u>ANOVA Summary Table</u>					
Source of Variation	DF	Sum of Squares	Mean Squares	F-Ratio	Significance Level
Replications	2	0.386	0.193		
A	13	18.744	1.442	7.699	0.000
Experimental Error	26	4.869	0.187	54.975	0.000
Sampling Error	42	0.143	0.003		
Total	83	24.143			

### **4. Two-Factor Factorial in Completely Randomized Design**

Syntax:

```
ANOVA (4) <Dependent variable> (<Factor A>) (<Factor B>)  
ANOVA (4) (<Factor A variable list>) (<Factor B>)  
ANOVA (4) (<Factor A>) (<Factor B variable list>)
```

Discussion: When compared to a one-factor-at-a-time approach, factorial designs are superior because they enable interactions between different factors to be explored. Instead of performing two experiments (one for each factor), the researcher can perform one experiment to determine the effects of each factor and their interaction. Unbalanced designs are acceptable.

An Example: Sixty baby male rats were randomly assigned to one of six feeding treatments. The dependent variable is the weight gain of the rats. The feeding treatments were a combination of two factors, source and level of protein. Three of the rats died before the experiment was completed. The six feeding treatments were a combination of two factors:

Factor A (3 levels): Source of protein: Beef, Cereal, Pork

Factor B (2 levels): Level of protein: High, Low

The analysis of variance table follows:

<u>ANOVA Summary Table</u>					
Source of Variation	DF	Sum of Squares	Mean Squares	F-Ratio	Significance Level
A	2	159.267	79.633	0.384	0.683
B	1	3716.946	3716.946	17.912	0.000
AB (Interaction)	2	1444.372	722.186	3.480	0.038
Error	51	10583.175	207.513		
Total	56	15903.760			

## **5. Two-Factor Factorial in Randomized Complete Block Design**

Syntax:

ANOVA (5) <Dep. var.> (<Factor A>) (<Factor B>) (<Factor C>)  
 ANOVA (5) (<Factor A variable list>) (<Factor B>) (<Factor C>)  
 ANOVA (5) (<Factor A>) (<Factor B variable list>) (<Factor C>)  
 ANOVA (5) (<Factor A>) (<Factor B>) (<Factor C variable list>)

Note: Factor A is always an experimental treatment

Factor B is always an experimental treatment

Factor C is always the replicate or block

Discussion: This ANOVA design is the same as the previous two-way factorial design except that replications have been added. This design will always contain exactly one observation per cell. It should be noted that for small experiments, the degrees of freedom for error may be quite small with this design.

An Example: Riboflavin content of collard leaves can be determined by a chemical technique known as fluorometric determination. An experiment was designed to study this technique. Factor A is the size of leaf used to make the determination (0.25 grams and 1.00 grams), and factor B is the effect of the inclusion of a permanganate-peroxide clarification step in the chemical process. The procedure was replicated on three successive days. There is one observation for each cell of the design. The dependent variable is apparent riboflavin concentration (mg./gm.) in collard leaves.

The analysis of variance table follows:

<u>ANOVA Summary Table</u>					
Source of Variation	DF	Sum of Squares	Mean Squares	F-Ratio	Significance Level
Replications	2	3.762	1.881		
A	1	36.401	36.401	4.450	0.079
B	1	716.108	716.108	87.547	0.000
AB (Interaction)	1	13.021	13.021	1.592	0.254
Error	6	49.078	8.180		
Total	11	818.369			

## **6. Three-Factor Factorial in Completely Randomized Design**

Syntax:

```
ANOVA (6) <Dep. var.> (<Factor A>) (<Factor B>) (<Factor C>)
ANOVA (6) (<Factor A variable list>) (<Factor B>) (<Factor C>)
ANOVA (6) (<Factor A>) (<Factor B variable list>) (<Factor C>)
ANOVA (6) (<Factor A>) (<Factor B>) (<Factor C variable list>)
```

Note:   Factor A is always an experimental treatment  
           Factor B is always an experimental treatment  
           Factor C is always an experimental treatment

Discussion: The three-way design is used when there are three factors to investigate. There are three main effects (A, B and C) and four interaction effects (AB, AC, BC and ABC). Unbalanced designs are acceptable. In the special case where there is only one observation per cell, the error term becomes equal to zero. In this case, it is usually assumed that the three-way interaction term is not significantly different from the error term, and the three-way interaction is used in place of the error term. With small numbers of levels for the factors, this can leave very few degrees of freedom for the error (3-way interaction) term.

An Example: A researcher was investigating the effect of various fertilizers on the growth of carrots. The three factors were nitrogen (N), potassium (K) and phosphorous (P). Three levels of concentration levels were tested for each factor (low, medium, and high). This resulted in 27 different fertilizer combinations. The dependent variable is the weight of the carrot root (in grams) grown under each of the fertilizer conditions. Note that in the special case of one observation per cell (i.e., one carrot for each fertilizer combination), no error term appears in the ANOVA table.

The analysis of variance table is as follows:

<u>ANOVA Summary Table</u>					
Source of Variation	DF	Sum of Squares	Mean Squares	F-Ratio	Significance Level
A	2	488.367	244.184	10.515	0.006
B	2	1090.656	545.328	23.483	0.000
C	2	49.148	24.574	1.058	0.391
AB (Interaction)	4	142.585	35.646	1.535	0.280
BC (Interaction)	4	592.624	148.156	6.380	0.013
AC (Interaction)	4	32.347	8.087	0.348	0.838
Error	8	185.776	23.222		
Total	26	2581.505			

### **7. Three-Factor Nested Design**

Syntax:

ANOVA (7) <Dep. var.> (<Factor A>) (<Factor B>) (<Factor C>)  
 ANOVA (7) (<Factor A variable list>) (<Factor B>) (<Factor C>)  
 ANOVA (7) (<Factor A>) (<Factor B variable list>) (<Factor C>)  
 ANOVA (7) (<Factor A>) (<Factor B>) (<Factor C variable list>)

Note: Factor A is always the experimental unit  
 Factor B is always a sub-unit of Factor A  
 Factor C is always a sub-unit of Factor B

Discussion: When each sample is composed of subsamples, we have a nested or hierarchical design. The objective of this design is to estimate the variance components associated with the various nested factors. Unbalanced designs are acceptable. In the special case of exactly one observation per cell, the error term is zero, and it will not be printed.

An Example: An investigator wanted to estimate calcium concentration in leaves of turnip plants. Four plants were taken at random and three leaves (samples) were randomly selected from each plant. Two subsamples were then taken from each leaf, and calcium was determined by microchemical methods. The objectives of the experiment were to estimate the variability in concentration across plants, between leaves of the same plant, and within subsamples of the same leaf.

The analysis of variance table follows:

<u>ANOVA Summary Table</u>					
Source of Variation	DF	Sum of Squares	Mean Squares	F-Ratio	Significance Level
A	3	7.560	2.520		
B within A	8	2.630	0.329		
C within B	12	0.060	0.007		
Total	23	10.270			

## **8. Split-Plot With Completely Randomized Design of Main Plots**

Syntax:

```
ANOVA (8) <Dep. var.> (<Factor A>) (<Factor B>) (<Factor C>)  
ANOVA (8) (<Factor A variable list>) (<Factor B>) (<Factor C>)  
ANOVA (8) (<Factor A>) (<Factor B variable list>) (<Factor C>)  
ANOVA (8) (<Factor A>) (<Factor B>) (<Factor C variable list>)
```

Note:   Factor A is always an experimental treatment  
          Factor B is always an experimental treatment  
          Factor C is always the replicate or block

Discussion: The term split-plot comes from agricultural experimentation. Split-plot designs contain two treatment factors. The main plots are the experimental units for one of the factors, and the subplots are the experimental units for the other factor. Split-plots are a repeated measure design.

An Example: In this experiment, six subjects were divided into two groups according to the method they were told to use for calibrating dials. Three subjects used method A1 to calibrate the dials, and three subjects used method A2. Each subject was told to calibrate four differently shaped dials (B1, B2, B3 and B4). The dependent variable is the accuracy of each calibration attempt. Factor A is the method of calibrating dials, and factor B is the shape of dials. The three subjects in each group are the replicates. This design will always contain exactly one observation per cell.

The analysis of variance table follows:

<u>ANOVA Summary Table</u>					
Source of Variation	DF	Sum of Squares	Mean Squares	F-Ratio	Significance Level
A	1	51.042	51.042	11.893	0.026
Error A	4	17.167	4.292		
B	3	47.458	15.819	12.798	0.000
AB	3	7.458	2.486	2.011	0.166
Error B	12	14.833	1.236		
Total	23	137.958			

## **9. Split-Plot With Randomized Complete Block Design of Main Plots**

Syntax:

```
ANOVA (9) <Dep. var.> (<Factor A>) (<Factor B>) (<Factor C>)  
ANOVA (9) (<Factor A variable list>) (<Factor B>) (<Factor C>)  
ANOVA (9) (<Factor A>) (<Factor B variable list>) (<Factor C>)  
ANOVA (9) (<Factor A>) (<Factor B>) (<Factor C variable list>)
```

Note: Factor A is always an experimental treatment  
Factor B is always an experimental treatment  
Factor C is always the replicate or block

Discussion: This design is preferable to the previous design when homogeneous blocks of experimental units are available, thus allowing a more accurate comparison of treatments by eliminating intra-block variability. Each AB treatment combination is replicated in each block. There will always be exactly one observation per cell with this design.

An Example: This example studies the effects of alfalfa variety and the date of harvest on yields. Six plots (replicates) were used. Factor A is alfalfa variety and factor B is the date of harvest. Factor C are the replicates (six plots).

The analysis of variance table follows:

<u>ANOVA Summary Table</u>					
Source of Variation	DF	Sum of Squares	Mean Squares	F-Ratio	Significance Level
A	2	0.178	0.089	0.653	0.541
Replications	5	4.150	0.830		
Error A	10	1.362	0.136		
B	3	1.962	0.654	23.390	0.000
AB (Interaction)	6	0.211	0.035	1.255	0.297
Error B	45	1.259	0.028		
Total	71	9.122			

## **10. Split-Plot With Sub-Unit Treatments Arranged in Strips**

Syntax:

ANOVA (10) <Dep. var.> (<Factor A>) (<Factor B>) (<Factor C>)  
ANOVA (10) (<Factor A variable list>) (<Factor B>) (<Factor C>)  
ANOVA (10) (<Factor A>) (<Factor B variable list>) (<Factor C>)  
ANOVA (10) (<Factor A>) (<Factor B>) (<Factor C variable list>)

Note: Factor A is always an experimental treatment  
Factor B is always an experimental treatment  
Factor C is always the replicate or block

Discussion: Instead of randomizing the sub-unit treatment independently within each unit, it is often necessary (or desirable) to have the sub-unit treatment arranged in strips across each replication. This design has an advantage over the previous design because it allows the determination of experimental error (error not attributable to either main factor). The layout may be particularly convenient for some field experiments. This design always contains exactly one observation per cell.

This design sacrifices precision on the main effects of A and B in order to provide higher precision on the interaction term which will generally be more accurately determined than in either randomized blocks or the simple split-plot design. For a 5 by 3 design, the appropriate arrangement (after randomization) might be as shown below for 2 replications:

Replication 1	Replication 2
a3 a1 a2 a0 a4	a1 a4 a0 a2 a3
b2	b1
b0	b2
b1	b0

An Example: The researcher used ten varieties and three generations of corn seed to study the effect of yield. The generations (a, b and c) appear in strips across blocks as well as the hybrid number.

The analysis of variance table follows:

<u>ANOVA Summary Table</u>					
Source of Variation	DF	Sum of Squares	Mean Squares	F-Ratio	Significance Level
A	9	77.683	8.631	0.939	0.524
Replications	1	2.817	2.817		
Error A	9	81.017	9.002		
B	2	35.433	17.717	2.183	0.314
Error B	2	16.233	8.117		
AB (Interaction)	18	61.567	3.420	2.627	0.024
Residual	18	23.433	1.302		
Total	59	298.183			

## 11. Latin Square Design

Syntax:

ANOVA (11) <Dep. var.> (<Factor A>) (<Factor B>) (<Factor C>)  
 ANOVA (11) (<Factor A variable list>) (<Factor B>) (<Factor C>)  
 ANOVA (11) (<Factor A>) (<Factor B variable list>) (<Factor C>)  
 ANOVA (11) (<Factor A>) (<Factor B>) (<Factor C variable list>)

Note: Factor A is always the row factor

Factor B is always the column factor

Factor C is always the treatment factor

Discussion: Latin square designs are very efficient when a small number of treatments are being tested because treatment comparisons are made more precise by eliminating row and column effects. The basic characteristic of a Latin square design is that each treatment appears once in each row and once in each column. With small numbers of treatments, there are few degrees of freedom for error. The limitation of the Latin square for a large number of treatments is due to the requirement that there



be the same number of replications as treatments. Thus, the most generally used Latin squares vary from 4 by 4 to 10 by 10. Latin square designs are also useful to study sequences of treatments and/or carry-over of treatments. This design will always contain exactly one observation per cell.

An Example: The field layout for 5 irrigation treatments (A, B, C, D and E) was as follows:

	Columns				
	1	2	3	4	5
Row 1	E	D	A	B	C
Row 2	C	E	D	A	B
Row 3	A	C	B	E	D
Row 4	D	B	E	C	A
Row 5	B	A	C	D	E

The analysis of variance table follows:

<u>ANOVA Summary Table</u>					
<u>Source of Variation</u>	<u>DF</u>	<u>Sum of Squares</u>	<u>Mean Squares</u>	<u>F-Ratio</u>	<u>Significance Level</u>
A	4	9.421	2.355	1.591	0.240
B	4	45.177	11.294	7.628	0.003
C	4	106.756	26.689	18.026	0.000
Error	12	17.767	1.481		
Total	24	179.120			

### ***Summary of Anova Designs***

Anova Type=1 One-Factor Completely Randomized Design

Factor A - experimental treatment

Balanced or unbalanced design

Anova Type=2 Randomized Complete Block Design

Factor A - experimental treatment

Factor B - replicates or blocks

Balanced design only

Anova Type=3 Two-Factor Factorial in Completely Randomized Design

Factor A - experimental treatment

Factor B - replicates or blocks

Factor C - sampling or determinations

Balanced design only

Anova Type=4 Two-Factor Factorial in Completely Randomized Design

Factor A - experimental treatment

Factor B - experimental treatment

Balanced and unbalanced design

Anova Type=5 Two-Factor Factorial in Randomized Complete Block Design

Factor A - experimental treatment

Factor B - experimental treatment

Factor C - replicates or blocks

Balanced design only

Anova Type=6 Three-Factor Factorial in Completely Randomized Design

Factor A - experimental treatment

Factor B - experimental treatment

Factor C - experimental treatment

Balanced or unbalanced design

Anova Type=7 Three-Factor Nested Design

Factor A - experimental unit

Factor B - sub-unit of factor A

Factor C - sub-unit of factor B

Balanced or unbalanced design

Anova Type=8 Split-Plot With Completely Randomized Design of Main Plots

Factor A - experimental treatment

Factor B - experimental treatment

Factor C - replicates or blocks

Balanced design only

Anova Type=9 Split-Plot With Randomized Complete Block of Main Plots

Factor A - experimental treatment

Factor B - experimental treatment

Factor C - replicates or blocks

Balanced design only

Anova Type=10 Split-Plot With Sub-Unit Treatments Arranged in Strips

Factor A - experimental treatment

Factor B - experimental treatment

Factor C - replicates or blocks

Balanced design only

Anova Type=11 Latin Square Design  
Factor A - row factor  
Factor B - column factor  
Factor C - experimental treatment  
Balanced design only

---

## CANONICAL Command

Canonical correlation analysis is a powerful multivariate statistical technique to study the intercorrelation structure of two sets of variables (variable set 1 and variable set 2). Each "set" of variables must contain at least two variables, although the number of variables in each set does not need to be the same. StatPac defines set 1 as the set with the larger number of variables, and set 2 as the set with the smaller number of variables. This convention is used to speed up the execution of the analysis.

Often (but not necessarily), one set is regarded as a "dependent" set and the other set is regarded as an "independent" set. For example, buying behavior could be a "dependent" variable set, and various personality characteristics of the buyers could be the "independent" variable set. Usually, the variables in each set represent two distinct variable "domains" that are conceptually different and measured on the same individuals.

Canonical correlation analysis is related to many other statistical techniques. Consider the following analogy. If each set of variables were to contain only one variable, simple correlation analysis would be equivalent to canonical correlation analysis. If one of the sets has one variable and the other set has two or more variables, then multiple regression analysis would be equivalent to canonical correlation analysis. Canonical correlation analysis can be seen as an extension of multiple regression when a single dependent variable is replaced by a set of dependent variables.

Canonical correlation analysis first calculates two new variables called the "first canonical variable pair". One of the new variables is calculated from set 1, and the other new variable is calculated from set 2. These new variables are constructed through linear combinations of the original variables so that the new canonical variable from set 1 has maximum correlation with the new canonical variable from set 2.

The second canonical variable pair is created in a similar fashion. One canonical variable is constructed from the first set of original variables and has maximum correlation with the second canonical variable created from the second set of original variables. This process is subject to the constraint that the second canonical variable pair is uncorrelated with the first canonical variable pair. In essence, the first canonical pair (one canonical variable from each set) is chosen so that the correlation between the two sets of original variables is maximized. The second set of canonical variables maximizes the "remaining correlation" between the two sets of variables (not picked up by the first set of canonical variables). The same philosophy applies to subsequent canonical variable pairs created.

This concept of independence of each canonical variable pair is similar to that involved in calculating principal components. The difference lies in the way the canonical variable pairs (or weights assigned to the original variables of each of the two sets) are created. In canonical correlation, the weights for each pair of canonical variables are calculated with the intent of maximizing "remaining" correlation between the two sets of variables, while in principal components the weights are

calculated with the intent of maximizing the remaining "correlation (or variance) structure" in one set of variables.

Canonical analysis provides a powerful method of studying the correlation structure between two sets of variables. For example, assume we are interested in studying the inter-correlation of variables from two different tests, each containing 20 variables. The correlation matrix would contain 400 different correlation coefficients to examine. This would be a very large task. The canonical correlation analysis simplifies this task by:

1. Determining the maximum correlation possible of any linear combination of the original variables in each set.
2. Deriving two sets of weighting coefficients (one for each of the two sets of variables) to arrive at a new composite variable (canonical variate) representing each set of variables. This provides a structured way of examining the contribution of each original variable (from each set). More specifically, this allows one to pinpoint groups of variables from each set that are highly correlated. Often, these groups can be given theoretical meaning.
3. Deriving additional linear functions which maximize the remaining correlation between the two sets, subject to being independent of the preceding set(s) of linear compounds.
4. Testing the statistical significance of the correlation measures, thereby determining the minimum number of linear functions required to account for the correlation structure of the two sets.
5. Assessing the overall extent of correlation between the two sets of variables.

For a detailed mathematical discussion of canonical correlation analysis (as well as the interpretation of the canonical correlation output of several examples), the user is referred to the Lohnes and Cooley (1971) book given in the bibliography.

The syntax of the command to run canonical correlation analysis is:

**CANONICAL <Variable list 1> WITH <Variable list 2>**

As an example, a researcher wished to study the correlation between 3 physiological measurements WEIGHT (V1), WAIST (V2) and PULSE (V3) with 3 exercise variables CHINS (V4), SITUPS (V5) and JUMPS (V6).

The command to run canonical correlation analysis could be specified in several ways:

**CANONICAL WEIGHT, WAIST, PULSE WITH CHINS, SITUPS,  
JUMPS**

**CANONICAL V1 V2 V3 WITH V4 V5 V6**

**CANONICAL V1-V3 WITH V4-V6**

**CA V1-V3 WITH V4-V6** (Note: CANONICAL can be abbreviated as CA)

The variable list can consist of variable labels and/or variable numbers. Either a comma or a space can be used to separate the variables from each other. The keyword WITH is used to separate the two sets of variables.

## Descriptive Statistics

The mean and standard deviations for all variables in both sets can be printed with the descriptive statistics option (DS=Y).

## Example of Descriptive Statistics

<u>Variables in the Analysis - Descriptive Statistics</u>				
Variable		Set	Mean	SD
V1	Weight (pounds)	1	178.600	24.691
V2	Waist (inches)	1	35.400	3.202
V3	PULSE (resting)	1	56.100	7.210
V4	CHINS	2	9.450	5.286
V5	SITUPS	2	145.550	62.567
V6	JUMPS	2	70.300	51.277
Number of Variables In Set 1 : 3				
Number of Variables In Set 2 : 3				
Number of Valid Cases = 20				
Number of Missing Cases = 0				
Response Percent = 100.0 %				

## Simple Correlation Matrix

The simple correlation matrix is printed with the SC=Y option.

## Example of a Simple Correlation Matrix

<u>Simple Correlation Matrix</u>						
		V1	V2	V3	V4	V5
V2	r	0.870				
	p	0.000				
V3	r	-0.366	-0.353			
	p	0.113	0.127			
V4	r	-0.390	-0.552	0.151		
	p	0.089	0.012	0.526		
V5	r	-0.493	-0.646	0.225	0.696	
	p	0.027	0.002	0.340	0.001	
V6	r	-0.226	-0.191	0.035	0.496	0.669
	p	0.337	0.419	0.884	0.026	0.001

### ***R-Squared Table***

This option (RS=Y) provides the multiple correlation (r-squared) analysis of each variable in each set, regressed on all variables in the other set. The r-squared value given in the first row, is the squared multiple correlation of the first variable in the first set, with the entire set of variables in the second set. The r-squared value in the last row is the squared multiple correlation of the last variable in the second set, with the entire set of variables in the first set, and so forth.

This output is a useful supplement to canonical correlation analysis, especially when used in conjunction with the redundancy analysis output described below. The output from this option allows one to measure whether certain variables are more correlated than other variables (within the same set) to the other set as a whole.

### ***Example of an R-Squared Table***

<u>R-Squared Table</u>			
Variable		IV's From Set	R-Squared
V1	Weight (pounds)	2	0.268
V2	Waist (inches)	2	0.548
V3	PULSE (resting)	2	0.075
V4	CHINS	1	0.340
V5	SITUPS	1	0.436
V6	JUMPS	1	0.054

### ***Standardized Coefficients***

Standardized coefficients for all canonical variables in both sets are printed with the option RC=Y. The standardized canonical variable coefficients are printed for the first set of variables, followed by those of the second set. These are the weights assigned to each original variable (standardized by subtracting the mean and dividing by the standard deviation) to construct the new variables (standardized canonical variable pairs) for each set.

### ***Example of a Standardized Coefficients Table***

<u>Standardized Coefficients for Canonical Variables</u>			
	CANO1	CANO2	CANO3
<u>Set 1</u>			
V1	-0.775	1.884	-0.191
V2	1.579	-1.181	0.506
V3	-0.059	0.231	1.051
<u>Set 2</u>			
V4	-0.349	0.376	-1.297
V5	-1.054	-0.123	1.237
V6	0.716	-1.062	-0.419

## Correlation of Canonical Pair with Original Variables

The canonical variable loadings are provided for each set in a separate table. These loadings are the correlation of the canonical variables with the original variables in the set and are similar to factor loadings in factor analysis. The loadings (correlations) therefore indicate the relative contribution of each original variable (in each set) in constructing the new canonical variables. This is the most useful table in understanding the correlation structure of the two sets of variables, and it provides insight into interpretation of the canonical variable pairs.

## Example of Canonical Variable Loadings

Canonical Variable Loadings for the First Set of Variables			
	CANO1	CANO2	CANO3
V1	0.621	0.772	-0.135
V2	0.925	0.378	-0.031
V3	-0.333	-0.041	0.942

Canonical Variable Loadings for the Second Set of Variables			
	CANO1	CANO2	CANO3
V4	-0.728	-0.237	-0.644
V5	-0.818	-0.573	0.054
V6	-0.162	-0.959	-0.234

## Redundancy Table

Canonical redundancy analysis examines 1) how well the original variables can be predicted from the canonical variables, and 2) how much of the "variance" of the two sets is in common (i.e., how much of the "variance" in one set can be accounted for by the other set). If  $RT=Y$ , a redundancy table is printed for each of the 2 sets of variables. Each table provides two statistics for each canonical variable.

The statistic given in the first column represents the proportion of the "variance" in the set of variables being analyzed that is accounted for by each canonical variable in that set. Since all canonical variables are independent, the total figure (last row) presents a measure of how well the canonical variables have been able to account for the "variance" in the original variables. It should be noted that, for the second set of variables, this total figure will always be 1 since there are as many new variables as original variables and the new variables are independent.

The statistic in the second column gives the proportion of the "variance" in the set that is explained by each canonical variable in the other set. This statistic is usually referred to as the "redundancy of a set given the other", and measures the extent of overlap between the two sets with respect to that canonical variable pair. A small redundancy value indicates that practically no overlap across sets is present in that particular pair of canonical variables. The total figure (last row of the second column) provides an overall measure of the extent to which one set is a good predictor of the other (based on the canonical variable pairs constructed). The redundancy measure is important because a very large canonical coefficient could be the result of a very large zero-order correlation of just one variable of one set with just one variable of the other set, and the remainder of the variables in the two sets could be essentially uninvolved. In this case, the canonical correlation for that pair of

canonical variables would be high but the redundancy for that pair of canonical variables would be low. The redundancy criteria would therefore be a better measure of the extent of overlap in the two "sets" than the measure provided by the canonical correlation associated with that canonical correlation pair.

### ***Example of a Redundancy Table***

<u>Redundancy Table for First Set of Variables, Given the Second Set</u>		
<u>Factor</u>	<u>Proportion of Total Variance Extracted</u>	<u>Redundancy of First Set, Given the Second Set</u>
CANO1	0.451	0.285
CANO2	0.247	0.010
CANO3	0.302	0.002
Total	1.000	0.297

<u>Redundancy Table for Second Set of Variables, Given the First Set</u>		
<u>Factor</u>	<u>Proportion of Total Variance Extracted</u>	<u>Redundancy of Second Set, Given the First Set</u>
CANO1	0.408	0.258
CANO2	0.434	0.017
CANO3	0.157	0.001
Total	1.000	0.277

### ***Rao's F-Statistic Table***

Various statistics, including Rao's F-statistic, are included in this table. It provides information to evaluate the extent of the correlation between the two sets of variables, the number of canonical variable pairs that are required to represent this correlation, and how successful the canonical variables have been in summarizing the interdependence of the two sets of variables.

Until recently, Rao's F-statistic table was virtually the only tool a researcher had to evaluate the above criteria. More recently, however, researchers have come to realize that these statistics can be misleading in the absence of the redundancy analysis table. The statistics in this table, combined with the redundancy table statistics, provide a more powerful approach to studying the performance of the canonical correlation analysis.

The canonical correlation given in the first row of this table (second column) is the correlation between the first pair of canonical variables (i.e., the correlation of the first new variable created from the first set of original variables with that of the first new variable created from the second set of variables). By construction, this is the maximum correlation that can be obtained between any linear combination of the first set of variables and any linear combinations of variables in the second set. The canonical correlation given in the second row of this table is the correlation between the second pair of canonical variables, and so forth.

The maximum number of canonical variable pairs that can be constructed is equal to the number of variables in the second set. Remember that, by construction, the *i*th canonical variable pair is independent of all the canonical variable pairs before it,



thus implying that the second canonical variable pair represents two new variables that are maximally correlated, subject to the condition that they are not picking out any of the correlation structure that was already picked up by a previous canonical variable pair.

The third column (canonical r-squared) is simply the square of the canonical correlation and represents the amount of variance in one canonical variate that is accounted for by its canonical variable counterpart in the other set. By definition, the canonical r-square given in the first row should be greater than the r-square value obtained by regressing only one variable from a set (say the first) with all the variables from the other set (the second set). This can be verified with the r-squared table (option RS). Should any of the values in the r-squared table be close to the first canonical r-square, it is doubtful that the other variables in that set are adding much to the correlation with the other set.

The fourth column provides Rao's F-statistic to test how many canonical variable pairs are needed to adequately represent the correlation structure of the two sets of variables. Some software uses Bartlett's chi-square statistic to test the hypothesis. StatPac provides Rao's F-statistic, as it is a more accurate test, especially with a small sample size.

The fifth and sixth columns give the degrees of freedom associated with this F-statistic, while the seventh column gives the significance level (probability value) associated with the F-statistic. The probability value in the first row provides a test of whether the canonical variables have accounted for a significant amount of the correlation structure between the two sets of variables (i.e., at least the first canonical correlation is significantly different from zero). The probability value given in the second row tests whether all the canonical correlations (except the first) are significantly different from zero. The probability value in the third row tests whether all canonical correlations (except the first two) are significantly different from zero, and so forth. StatPac provides output on all canonical variable pairs possible; it is up to the user to determine (based on magnitude of the canonical correlations as well as the significance level of the F-statistics) how many of these pairs are relevant. The last column provides Wilk's lambda as each of the canonical variables is added; this criterion is used in calculating the F-statistic.

### ***Example of a Rao's F-Statistic Table***

<u>Rao's F-Statistic with Successive Roots Removed</u>						
<u>Roots Removed</u>	<u>Canonical Correlation</u>	<u>Canonical R-Squared</u>	<u>Rao's F-Statistic</u>	<u>DF</u>	<u>Prob.</u>	<u>Wilk's Lambda</u>
0	0.796	0.633	2.048	9 & 34.2	0.064	0.350
1	0.201	0.040	0.176	4 & 30.0	0.949	0.955
2	0.073	0.005	0.085	1 & 16.0	0.775	0.995

### ***Mean Substitution***

Mean substitution is one method often used to reduce the problem of missing information. Sometimes, canonical correlation analysis is difficult because if one variable is not known, the whole record must be excluded from the analysis. This could substantially reduce the number of records that are included in the analysis, especially when there are many variables in the analysis. Mean substitution overcomes this problem by replacing any missing variable with the mean of that variable.

## Save Canonical Pair

Researchers often want to save the first two canonical variables for future analysis. You can save them with the options command SP=Y. At the conclusion of the analysis you will be given the opportunity to merge the new variables into the original data.

## Labeling and Spacing Options

Option	Code	Function
Labeling	LB	Sets the labeling for descriptive statistics to print the variable label (LB=E), the variable name (LB=N), or the variable number (LB=C).
Column Width	CW	Sets the minimum width of the columns (in inches).
Column Spacing	CS	Sets the spacing (in inches) between the columns.
Decimal Places	DP	Sets the number of decimal digits that will be shown.

---

## MAP Command

Perceptual mapping refers to a broad range of market research techniques to study consumer perceptions of products ("brands") in a class based on the product attributes. To achieve this objective, an attempt is made to reduce the dimensionality of the product/attribute space; plots (perceptual maps) are then used to graphically display consumer perceptions of "brands" in a category.

Marketing researchers use perceptual mapping to identify new opportunities and marketing strategies. One of its primary application is to analyze the effectiveness of marketing campaigns designed to modify people's perceptions of a product.

The four most frequently used attribute-based statistical tools for perceptual mapping are discriminant analysis, principal components analysis, factor analysis and correspondence analysis.

The MAP command uses the correspondence analysis approach to perceptual mapping. Correspondence analysis is an exploratory statistical technique for examining multivariate categorical data (usually in the form of a banners table) and graphically displaying both the rows and columns of the table in the same multidimensional space. The object of correspondence analysis is to graphically summarize the correlations among row and column categories.

In marketing applications, columns are usually brands, and rows are image attributes. However, since the row and column categories are treated equally, the axes of the banners table could be swapped and the results would be the same. Correspondence analysis produces perceptual maps showing how consumers perceive various brands in relation to a set of image attributes. Brands with similar image profiles appear closer together on the map.

Correspondence analysis is synonymous to the following techniques: dual scaling, method of reciprocal averages, optimal scaling, pick-any scaling, canonical analysis of contingency tables, categorical discriminant analysis, homogeneity analysis, quantification of qualitative data.

Correspondence analysis starts with the usual banners (crosstabs) table. The table always contains one row variable and one column variable. These are referred to as the active variables.

The analysis begins by calculating the usual (contingency table) chi-square statistic, or more specifically, the chi-square statistic divided by sample size (Pearson's mean-squared contingency coefficient). This quantity is often referred to as total inertia. A large chi-square statistic implies association between rows and columns (i.e., lack of independence between rows and columns).

The purpose of correspondence analysis is to plot the nature of this association in a reduced dimensionality. It is similar to principal components in that new components (or factors) are extracted, each being independent of each other. If the first few components account for most of the "association in the table", we have been successful in reducing the dimensionality of the rows and columns.

The difference between principal components and correspondence analysis is the "variability" we are trying to explain. In principal components analysis, we are trying to explain the "correlation" or "covariance" structure of a set of variables, while in correspondence analysis we are trying to explain the "association" between rows and columns of a table of frequencies.

The decomposition of the chi-square proceeds by calculating two factors or vectors (one for the column levels and the other for the row levels). These factors are extracted such that the association between them is maximized (i.e., explains as much of the chi-square statistic as possible). Next, a second pair of factors is extracted. Their association is maximized to explain as much of the remaining chi-square statistic (subject to the constraint that this second factor pair is uncorrelated with the first factor pair). The same philosophy applies to subsequent factor pairs created.

This concept of independence of each factor pair is similar to the calculation of principal components. The difference lies in the way the factor pairs (or weights assigned to the levels of each row and column) are created. In correspondence analysis, the weights for each factor pair are calculated with the intent of maximizing "remaining" association between row and column levels, while in principal components the weights are calculated with the intent of maximizing the remaining "correlation (or variance) structure" in the variables.

The syntax of the command to run correspondence analysis is:

**MAP <Active row variable> BY <Active column variable>**

The only restriction on the data is that both variables must be categorical (alpha or numeric), and there must be at least three categories on each axis. However, since the row and column categories are treated equally, both can be either objects (brands) or attributes.

A simple example might be to map the relationship between purchasing DECISION (V1) and INCOME (V2). The command might be expressed in many ways:

**MAP INCOME BY DECISION**

**MAP DECISION BY INCOME**

**MAP V1 BY V2**

**MA V2 BY V1**

(Note: MAP may be abbreviated as MA)

## ***Passive Variables***

StatPac can perform multiple correspondence analysis. This means that passive rows and columns (often referred to as supplemental rows and columns) can be added to the perceptual maps. These are not included in the actual analysis, but rather, are superimposed upon the perceptual map of the active variables. Passive rows and columns are requested by adding them to the Map command. To include passive variables, the syntax for the Map command becomes:

**MAP <Active row variable><Passive row variables> BY <Active column variable><Passive column variables>**

Passive rows and columns are often used as reference points for active points in the plot. For example, if the active columns was brands, an example of a passive column point could be a hypothetical brand or a brand from a previous similar study.

## ***Stacking Variables***

Correspondence analysis always uses one active row variable and one active column variable (each containing at least three categories). It is often desirable, however, to perform correspondence analysis on contingency tables that contain more than two dimensions. Stacking variables allows you to combine two or more variables into a single variable. The Stack command may be used to create a new variable that represents all possible combinations of two or more other variables.

The following example would create a single variable called DEMOGRAPHICS, and it would be used as the active column variable. The number of categories in the new stacked variable will be the product of the number of categories in each of the individual variables.

**STACK DEMOGRAPHICS = AGE SEX  
MAP DEMOGRAPHICS BY BRAND**

If AGE had five categories and SEX had two categories, the resultant column variable (DEMOGRAPHICS) would contain ten categories. This would cover all the possible combinations of categories. Stacking variables should be used carefully because of the potential for a huge number of rows or columns. The maximum number of rows or columns is 150.

## ***Count/Percent Table***

Correspondence analysis always begins with a banners table. The banner table itself is called the count/percent table. It may be printed with the CP=Y option or suppressed with the CP=N option. The options to control the appearance of the Count/Percent table are identical to the options for the Banner command.

In the following example, the active row is Party Affiliation and the active column is Attitude on the New Government Proposal. Annual Income is a passive row, and Liberal/Conservative is a passive column.

### Example of a Count/Percent Table

N=83 Number Col%	Attitude On The New Government Proposal			Liberal Or Conservative	
	Agree	Neutral	Disagree	Conservative	Liberal
<u>Party Affiliation</u>					
Democrat	12 57.1%	2 12.5%	17 50.0%	12 37.5%	16 44.4%
Republican	7 33.3%	2 12.5%	7 20.6%	5 15.6%	8 22.2%
Independent	2 9.5%	12 75.0%	10 29.4%	15 46.9%	9 25.0%
<u>Annual Income</u>					
21,000 - 30,000	1 4.8%	8 50.0%	0 0.0%	3 9.4%	6 16.7%
30,000 - 40,000	2 9.5%	7 43.8%	14 41.2%	10 31.3%	13 36.1%
40,000 - 50,000	12 57.1%	1 6.3%	18 52.9%	19 59.4%	9 25.0%
Over 50,000	3 14.3%	0 0.0%	2 5.9%	0 0.0%	5 13.9%

### Design Summary for Abbreviated Labeling on Map

Perceptual maps can easily become overly cluttered, it is sometimes necessary to abbreviate the value labels with just a number. The design summary will contain the "plot number" for each row and column variable (i.e., a numeric abbreviation that can be used when creating large perceptual maps). The AB=Y option may be used to abbreviate the map so it uses numbers instead of value labels. A design summary will then be printed that prints the value labels for all active and passive rows and columns. This feature is only important when there are many rows and columns in the banners table.

### Example of a Design Summary

<u>Summary of Row and Column Labels for Plots</u>		
<u>Row/Column</u>	<u>Plot No.</u>	<u>Value Label</u>
Active Rows	1	Democrat
	2	Republican
	3	Independent
Active Columns	4	Agree
	5	Neutral
	6	Disagree
Passive Rows	7	21,000 - 30,000
	8	30,000 - 40,000
	9	40,000 - 50,000
Passive Columns	10	Over 50,000
	11	Conservative
	12	Liberal

### Correspondence Analysis Eigenvalue Summary

The correspondence analysis eigenvalue summary table is always included in the output. Each non-trivial eigenvalue represents a dimension (factor pair). The maximum number of eigenvalues for any contingency table is one less than the minimum of the number of rows and the number of columns in the contingency table. The sum of all eigenvalues equals the chi-square statistic of independence divided by sample size (total inertia).

The eigenvectors of the non-trivial eigenvalues define the coordinates of the factor pairs. If the first few eigenvalues are large relative to the remaining eigenvalues, it is then possible to display the association between rows and columns in a one or two-dimensional table. The eigenvalue table allows us to determine how much "information" is lost by ignoring all dimensions except for the first (or first and second). The first two dimensions should account for nearly 100% of the total row and column association (i.e., they completely explain all the association between the rows and columns).

### Example of Correspondence Analysis Eigenvalue Summary Table

<u>Correspondence Analysis Eigenvalue Summary</u>			
	<u>Eigenvalue</u>	<u>Proportion</u>	<u>Cumulative</u>
FACT1	0.254	97.44	97.44
FACT2	0.007	2.56	100.00
Total variance = 0.260			

### Summary of Row and Column Points

The correspondence analysis summary of row and column points table presents useful insight into the nature of the association between rows and columns, and is

printed for each analysis. This table provides information on the first two factor pairs (i.e., the first 2 dimensions) only.

### Example of Summary of Row and Column Points Table

<u>Correspondence Analysis Summary of Row and Column Points with Carroll-Green-Schaffer Scaling</u>				
<u>Factor 1</u>				
Value	Label	Coordinate	Corr.	Contr.
C1	Agree	-1.238	0.510	30.144
C2	Neutral	2.131	0.909	68.030
C3	Disagree	-0.238	0.048	1.806
R1	Democrat	-0.938	0.535	25.524
R2	Republican	-0.753	0.146	8.501
R3	Independent	1.713	0.998	65.975
<u>Factor 2</u>				
Value	Label	Coordinate	Corr.	Contr.
C1	Agree	-1.214	0.490	40.278
C2	Neutral	-0.672	0.091	9.415
C3	Disagree	1.066	0.952	50.307
R1	Democrat	0.874	0.465	30.814
R2	Republican	-1.819	0.854	68.963
R3	Independent	0.084	0.002	0.223

The first column (Coordinate) gives the factor scores for the first dimension. These are the coordinates that are used to plot the first dimension or axis. By definition, these coordinates are calculated to account for as much of the association in the contingency table as possible. Row and column levels with coordinates close to zero do not account for any of the association explained by the first dimension; they may however be important in the second dimension or axis.

The first correlation (Corr.) column gives the correlation of each row and column with the first dimension. A high value implies that the particular row (column) is important in describing the first dimension.

The first contribution (Contr.) column gives the row and column contribution to the association "picked-up" by the first dimension (expressed as a percentage). A row or column with a high marginal has a larger contribution than a row or column with a low marginal. It is important to note the difference between the concepts measured by the *Corr.* and *Contr.* columns. *Corr.* simply measures the correlation of the rows and columns with the first dimension. It is therefore useful in defining the first dimension. *Contr.*, on the other hand, provides a measure of how useful a particular row or column is in explaining the contingency table association for the first dimension. A row or column may be highly correlated with the first dimension but explain very little of the association between rows and columns.

The *Contr.* column is useful in locating outliers. When a row or column has a very large absolute contribution and a large coordinate, it can be considered an outlier and has a major role in determining the first and/or second coordinates. Thought should be given to redefining this outlier point as a passive (supplementary) point and performing the analysis again without this point, thereby eliminating that point's influence in the creation of the first few axis. The point can be superimposed on the axis calculated for the remaining points.

The second column of coordinates gives the factor scores for the second dimension. These are the coordinates that are used to plot the second dimension or axis. By definition, these coordinates are calculated to account for as much of the association in the contingency table not accounted for by the first dimension. The second Corr. column gives the correlation of each row and column level with the second dimension. A high value implies that the particular row (column) is important in describing the second dimension. The second Contr. column is the row's (and column's) contribution to the association "picked-up" by the second dimension (expressed as a percentage). If a row or column has low contribution on both the first and second dimension, this can be due to one or more of the following reasons:

1. There is no association between that row (column) and the levels of columns (rows).
2. The third (and higher) dimensions account for a significant portion of the association in the contingency table and cannot be ignored.
3. The row or column marginal is small and therefore has only a small effect on the chi-square statistic.

### ***Carroll-Green-Shaffer Scaling of Coordinates***

The option CG=Y requests that the Carroll-Green-Shaffer scaling of coordinates be used. If CG=N, the usual "French school" correspondence analysis coordinates are calculated.

These "French school" coordinates do not allow one to compare distances between column and row points but rather only distance between column points or between row points. Carroll, Green and Shaffer claim that their method of scaling of coordinates allow one to compare both within and across group (row/column) distances. If eigenvalues are almost equal, the CGS and "French school" plots look almost the same, but if the first eigenvalue is considerably greater than the second, the plots can look very different.

### ***Plots***

The correspondence analysis plots show the relationship between a row and all columns or between a column and all rows. When using the "French school" technique (CG=N), distances between a row and a column point cannot be interpreted. When using the Carroll-Green-Schaffer coordinates, distances between a row and a column point can be examined.

Four types of maps can be created with the PL option.

- A. two-dimension row-column plot
- B. one-dimension row-column plot
- C. two-dimension row plot
- D. two-dimension column plot

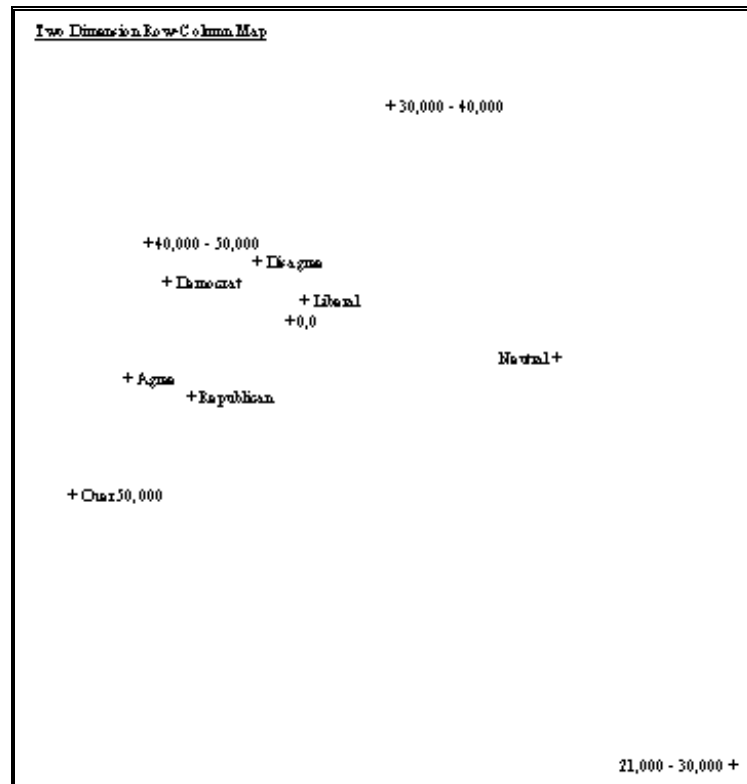
For example, the following option would create a two-dimension row-column plot, a two-dimension row plot, and a two-dimension column plot.

**OPTIONS PL=ACD**



The two-dimension row-column plot will show both row and column variables using the first two dimensions. If the first two eigenvalues (dimensions) account for a large portion of the total chi-square, then this plot provides a useful summary of the association between row and column points.

### ***Example of a Two-Dimension Row-Column Map***



In general, the more the row and column points are spread out, the higher the association and hence the higher the chi-squared statistic. Columns and rows represented by points in the plot relatively far from the origin account for relatively large portions of the lack of independence in the contingency table. Rows positioned close together in the map have similar profiles across the columns. Analogously, columns positioned close together in the plot exhibit similar profiles across the rows.

The two-dimension row plot is especially useful when there are a lot of row and column points to plot and/or one wishes to examine the distance between row points only.

The two-dimension column plot is especially useful when there are a lot of row and column points to plot and/or one wishes to examine the distance between column points only.

The one-dimension row-column plot is especially useful when the first dimension explains most of the association in the table. In this case the value of the first eigenvalue would be close to Pearson's mean-square contingency coefficient.

---

## Advanced Analyses Bibliography

The following bibliography is recommended for more detailed information on the advanced analyses module of StatPac for Windows:

### **Regression Analysis**

Draper N.R. and H. Smith - Applied Regression Analysis, 2nd Ed., New York, John Wiley and Sons, Inc., (1981).

Henderson, H.V. and P.F. Velleman, (1981) - Building Multiple Regression Models Interactively. Biometrics, Vol. 37, pp. 391-411.

Jenrich, R.I. - Stepwise Regression in Statistical Methods for Digital Computers, edited by Kurt Enstein, Anthony Ralston, Herbert S. Wiff, (1977).

Lewis, C.D. - Industrial and Business Forecasting Methods. Boston, Buterworth Scientific, (1982).

### **Probit and Logistic Regression Analysis**

Gunderson, M., (1974) - Retention of Trainees - A Study with Dichotomous Dependent Variables. Journal of Econometrics, Vol. 2, pp. 79-93.

Tobin, J. - The Application of Multivariate Probit Analysis to Economic Survey Data. Cowles Foundation Discussion Paper No. 1, December (1955).

Walker, S.H. and Duncan, D.B., (1967) - Estimation of the Probability of an Event as a Function of Several Independent Variables. Biometrika, Vol. 54, pp. 167-179.

Principal Components, Factor and Multicollinearity Analysis

Afifi, A.A. and Azen, S.P. - Statistical Analysis: A Computer Oriented Approach. New York, Academic Press, Inc. (1972).

Belsley, D.A., Kuh, E. and Welsch, R.E - Regression Diagnostics: Identifying Influential Data and Sources of Collinearity. New York, John Wiley and Sons, Inc. (1980).

Hocking, R.R. and Pendleton, O.J., (1983) - The Regression Dilemma. Communications in Statistics, Vol. A12, pp. 497-527.

Kim, Jae-On and C.W. Mueller - Factor Analysis: Statistical Methods and Practical Issues. California, Sage Publications, Inc. (1978).

Kramer, C.Y., (1978) - An Overview of Multivariate Analysis. Journal of Dairy Science, Vol. 61, pp. 848-854.

Veldman, D.J. - Fortran Programming for the Behavioral Sciences. New York, Holt, Rinehart and Winston, Inc. (1967).

### **Cluster Analysis**

Anderberg, M.R.- Cluster Analysis for Applications. New York, Academic Press (1973).

- Milligan, G.W., (1980) - An Examination of the Effect of Six Types of Error Perturbation on Fifteen Clustering Algorithms. *Psychometrika*, Vol. 45, pp. 325-342.
- Pung, G. and Stewart, D.W., (1983) - Cluster Analysis in Marketing Research: Review and Suggestions for Applications. *Journal of Marketing Research*, Vol. 20 (May), pp. 134-148.
- Spath, H. - Cluster Analysis Algorithms. Chichester, England, Ellis Horwood (1980).

### **Analysis of Variance**

- Cochran, W.G. and G.M. Cox - Experimental Designs, 2nd Edition, Wiley Books, (1957).
- Finney, D.J. - Experimental Design and its Statistical Basis, The University of Chicago Press, (1955).
- Quenouille, M.H. - The Design and Analysis of Experiments, Charles Griffin and Co. Ltd., (1953).
- Snedecor, G.W. and W. G. Cochran - Statistical Methods. Sixth Edition, The Iowa State Press, (1967).
- Steel, R.G.D. and J.H. Torrie - Principles and Procedures of Statistics, McGraw-Hill Book Company, Inc., (1960).
- Winer, B.J. - Statistical Principles in Experimental Design, McGraw-Hill Book Company, Inc., (1962).

### **Canonical Correlation Analysis**

- Cooley, W.W. and Lohnes, P.R. - Multivariate Data Analysis (Chapter 6). New York, John Wiley and Sons (1971).
- Green, P.E., Halbert, M.H. and Robinson, P.J., (1966) - Canonical Analysis: An Exposition and Illustrative Application. *Journal of Marketing Research*, Vol. 3, pp. 32-39.

### **Correspondance Analysis**

- Carroll-Green-Shaffer, (1986) - *Journal of Marketing Research*, Vol. 23, pp. 271-280.
- Dillion, William R., Frederick, Donald G., Tangpanichdee, Vanchai (1982) - A Note on Accounting for Sources of Variation in Perceptual Maps. *Journal of Marketing Research*, Vol. 19 (August), pp. 302-311.
- Fox, Richard J., (1988) - Perceptual Mapping Using the Basic Structure Matrix Decomposition. *Journal of the Academy of Marketing Science*, Vol. 16, pp. 47-59.
- Greenacre, M.J., - Theory and Applications of Correspondance Analysis. New York, Academic Press (1984).
- Greenacre, M.J. (1989) - *Journal of Marketing Research*, Vol. 26, pp. 358-368.

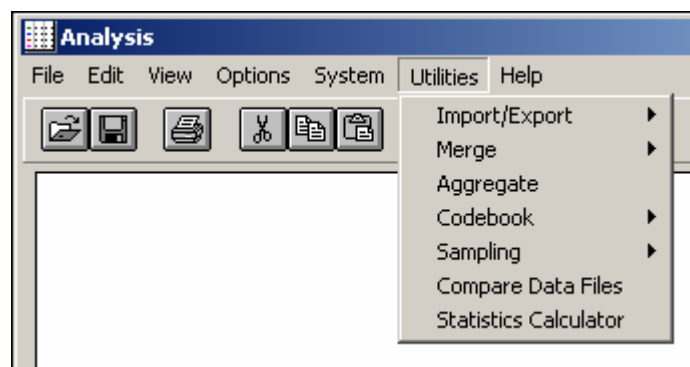


# Utilities

---

## Utility Programs

Seven utility programs are provided to give greater control and more versatility over studies and data files. They can be run from the Analysis, Utilities menu.



The Import and Export program will allow you to read files created by other software or write files that can be read by other software. Several formats are supported: Access, Excel, dBase, FoxPro, all prior versions of StatPac, all prior versions of StatPac Gold, Lotus, comma delimited, tab delimited, multiple record files, Internet response files, and plain-text e-mail.

The Merge program is used to merge variables from different studies and data files or to rearrange the sequence of variables in a file. It can merge data from up to five individual data files. It can also be used to concatenate (join) data files using the same codebook.

The Aggregate program is used to create a true or compositional aggregate study and data file. Aggregate files are useful for summarizing subgroups of data.

The Codebook program is used to quickly create a codebook, or to check a codebook and data file for errors. The Check program is used when you suspect that there is a problem in the codebook or data file. If a specific procedure won't run, this program can sometimes provide a solution. A common use of this program is when you are planning to use a data file created from a source other than StatPac, and you want to make sure that your study design matches the data file.

The Sampling program is used to generate a random number table, create a random digit dialing table for telephone studies, and to select a random sample from a data file.

The Compare Data Files program is used to compare two data files for differences. It is used to check the accuracy of data entry when a double entry system has been used.

The Statistics Calculator is used to calculate distributions, probabilities and other statistics from proportions and summary data.

There are also two additional programs that are not in the utility menu.

There is a program to swap IP with ID in the .asc file before doing an import. Some bulk email programs use a query string that far exceeds the 16 character ID that StatPac expects. It is often over a hundred random characters added as tracking by the bulk email program. StatPac uses the ID to combine multiple page surveys. A solution is to use the IP address as the ID number. The program Swap\_ID-IP.exe is not built into StatPac. It is located in the StatPac installation folder and can be run by double clicking on it with Windows Explorer. Run the program before importing the .asc file into StatPac. This program will work only if each respondent has a unique IP address. For business surveys, be careful. Respondents from the same company usually have the same IP address.

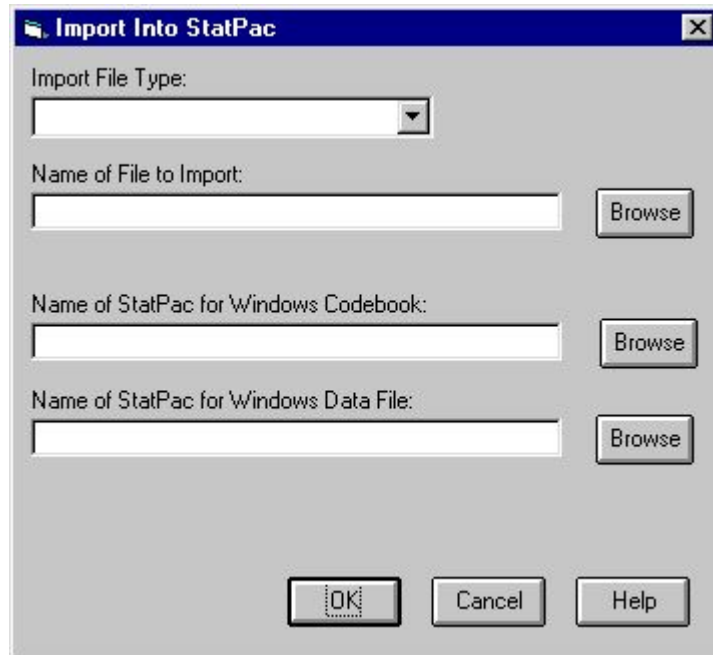
There is another program to fix .asc files that were downloaded in binary format instead of ASCII. This can only happen if you are using your own server and FTP program. Unix/Linux systems end each line with a carriage return. Windows ends each line with a linefeed and carriage return. When downloading in ASCII mode, the linefeed characters are inserted, but when downloading in the binary mode, the exact file contents are downloaded (no linefeed characters). This program will insert the linefeed characters in the file so it will be in Windows format. The program Fix\_Binary is not built into StatPac. It is located in the StatPac installation folder and can be run by double clicking on it with Windows Explorer.

---

## Import and Export

StatPac can import and export information to other software. Select Utilities, and then select import or export. Import means you want to convert a non-StatPac file into StatPac for Windows format. Export means you want to convert a StatPac for Windows file to a different format. When importing or exporting data, the original file(s) will remain intact and a new file(s) will be created.

When importing, select the type of file and name of the file to be imported. If you are importing from previous versions of StatPac, it will be assumed that the codebook/study name and the data file name are the same. The names for the StatPac for Windows codebook and data may be different.



Each of the import and export file formats is explained below.

## StatPac and Prior Versions of StatPac Gold

Prior versions of StatPac and StatPac Gold used a "codebook" file or "study design" files to store variable format and labels information. StatPac for Windows stores all this information in a codebook file. Because older versions of StatPac have limited labeling space, some labels may be truncated when exporting to an older version of StatPac.

The import program assumes that the data file name is the same as the study file name for the previous version. Thus, the "Name of the File to Import" will indirectly also specify the data file name. For example, if the file to import is SURVEY.EZ0, the program will try to import a data file called SURVEY.DAT from the same folder. If the data file does not exist, only the codebook will be imported. StatPac for Windows stores data in the same format as prior versions of StatPac (fixed format sequential ASCII). Therefore, if there is not a matching data file name, you can simply copy the old data file to the folder where you imported the codebook and use it without modification.

## Access, Excel, Paradox, dBase, and Lotus

StatPac can import or export to a variety common data base/worksheet formats. The appropriate extension will be used when you select the type of data base to be imported or exported. When importing to StatPac, the import procedure will create a new StatPac codebook and data file. Do not create a StatPac codebook before doing the import or it will be overwritten by the import procedure.

When importing from Lotus to StatPac, the default variable names are the worksheet column letters (e.g. Column A, Column B, etc.). If your worksheet contains locked column headings, they will be used as variable names for those columns. The column headings may be locked in Lotus by using the /Worksheet Titles Horizontal command. If the column titles are not locked, they will be written as the first record

in the StatPac data file (an undesirable situation). Also, be sure your worksheet does not contain any empty rows or dividing lines between the titles and the first row of data.

## DIF Files

DIF files were originally developed to provide a universal method of transferring data files from one program to another. Unfortunately, many program developers decided to add their own special features to the DIF file. The result was that that "universal" compatibility of DIF files became marginal.

## Comma Delimited and Tab Delimited Files

StatPac can import and export comma and tab delimited files. There are many software packages that can interchange data in this format. A comma delimited file is a sequential ASCII file where the variables are separated from each other by commas (rather than each variable using a fixed number of characters). In a tab delimited file, the separator is a tab character. Tab delimited imports and exports are generally more reliable than comma delimited files.

When importing, StatPac creates a new codebook based on the field widths required to hold the data. If a codebook already exists, you may use it instead of creating a new codebook. If a data file already exists, you'll be offered the option of appending to the existing data file or deleting it. Selecting append will add the newly imported data to the end of the existing data file.

Many software packages write quotation marks around alpha fields, while others do not. When importing a comma delimited file, all quotation marks will automatically be eliminated, since StatPac does not use quotes. When exporting to a comma delimited file, any field containing a comma will always be enclosed in quotes. The StatPac.ini file contains a setting *QuoteAlphaFields*. When *QuoteAlphaFields=1*, all alpha fields will be quoted when exporting to comma delimited. When set to zero, only fields containing a comma will be enclosed in quotes.

Many software packages can read or write a header record in comma and tab delimited files. A header record is usually the first record in the data file. It contains the names of the variables instead of actual data. If you are importing a comma or tab delimited file and don't know if there is a header record, load the file into your word processor and look at it. If the first line in the file is the names of the variables, there is a header record. If the first record looks just like the other records, it's data, and not a header record.

When exporting to a tab or comma delimited file, StatPac will give the option to convert the raw data to the value labels. For example, if the first variable were gender and coded 1=Male and 2=Female, a normal export would write a 1 or 2 for the variable. If the *Expand To Text* option is selected, it would write Male or Female to each data record instead of the raw data.

When exporting to a tab or comma delimited file (or Excel), with the intention of being able to import the file into SPSS, StatPac will give the option to also create a SPSS syntax file. This file is a text file with the same name as the exported file except it will have a .sps extension. In SPSS, first import the data. Then load the .sps file into Notepad or another text editor. Copy the contents of the file into the SPSS syntax editor and click play. This will create the variable and value labels in SPSS. When using this feature, StatPac may modify (abbreviate) the variable names, labels and value labels in order to fit the limited space offered by SPSS.



The tab delimited import utility can be used to import a text file for Verbatim Blaster open-ended response coding. For example, you might have used Microsoft Word to enter verbatim comments into a .txt file. Each person's comments were entered as a paragraph (i.e., a continuous string of text ending with a carriage return). This file can be imported as a tab delimited file. Since there are actually no tabs in the file, StatPac will correctly import the text into a codebook and data file containing a single variable. The variable will be an alpha type and will be as long as necessary to hold the open-ended comments.

Exported tab delimited files may use a .txt or .tsv (tab separated variables) extension. Exported comma delimited files may use a .txt or .csv (comma separated variables) extensions

When exporting to a tab or comma delimited file, keep in mind that many programs (Excel, Access, etc.) limit the number of columns to 255 (while StatPac can have as many as 2,000). If your codebook has more than 255 variables, an export to an Access file is preferred because it will split the data into multiple tables as necessary. Otherwise, you'll have to use the Write command to create a series of codebooks and data files (each containing 255 or fewer variables) and then export each one individually to a delimited file.

## Files Containing Multiple Data Records per Case

Many researchers want to use data that is in card-image format on a mainframe computer. Also, many data entry services are capable of only punching data in card-image format. While it is relatively easy to download data from a mainframe, it often comes in 80-column format. If there is only one record per case, this data can be read by StatPac without performing an import. However, when there is more than one "card image" (i.e., record) per case, it becomes necessary to concatenate (join) the "card-image" records together to produce a StatPac readable file.

Importing a multiple record file that looks like this...

```
Card 1 Case 1
Card 2 Case 1
Card 3 Case 1
Card 1 Case 2
Card 2 Case 2
Card 3 Case 2
etc.
```

will become a StatPac file that looks like this...

```
Card 1 Case 1 Card 2 Case 1 Card 3 Case 1
Card 1 Case 2 Card 2 Case 2 Card 3 Case 2
etc.
```

StatPac requires that a data record be a continuous stream of characters terminated with a carriage return and linefeed. This program will read a file in multiple record format and create a new data file with one record per case. The filename should have a .txt extension.

StatPac assumes that there are 80 characters in each record of the multiple record file. If the "card-image" record length is less than 80, StatPac will pad the records with spaces before combining them

You will need to specify how many records there are for each case. If the downloaded data file has 3 records per case, you will answer 3 (even if the third "card-image" record is only partially used).

## Internet Files

The preferred method of performing Internet surveys is to store the responses in a file on the server. When using the method, responses are stored in ASCII (.asc) text format. When you're ready to perform an analysis, download the file to your local computer using an FTP program or Auto Transfer. If you use a different FTP program, be sure to set it to download the file as an ASCII (not binary) file.

If you use Auto Transfer, the downloaded file will automatically be imported into StatPac. If you manually download the file, you will need to use this import utility to convert the .asc file to StatPac data.

Downloaded Internet response files are not automatically deleted from your server. Therefore, each time you download the responses, it will be the entire set of responses since the beginning of the survey. StatPac will offer you the choice of deleting the existing data file or appending to the end of it. Since the downloaded file is usually the entire data set, you would normally want to replace the existing data file with the newly downloaded data.

## Excel Files

Excel is probably the most popular format for exchanging data among programs. You can import and export .xls (Excel 2003) and .xlsx files (Excel 2007-2010). The default extension for Excel exports is .xls. You can explicitly change the file name extension to .xlsx during the export, or you can change the default by editing the StatPac.ini system defaults file. (Set ExcelExportExtension=.xlsx). The first row in the worksheet will be imported as the variable names.

## Email Surveys

Because of the variety of Email programs, it is not possible to describe the exact steps you must take to import a returned Email survey. Each Email program operates a little differently, and you will need to experiment with your program.

StatPac provides import capabilities for CGI and plain text Email surveys. CGI Email would be produced by a survey placed on a web site that used StatPac's email method of capturing responses. A plain-text survey would be produced by a survey that was simply part of the text in the body of an Email.

Select e-mail as the import type and use the browse button to select the file to be imported. Usually, this would be a .mbx file (i.e., a mailbox in Outlook Express or Eudora where the e-mails were filtered to). Use the browse button to select the existing StatPac codebook and specify the name of the data file. If the data file does not exist, it will be created by the import procedure. If it does exist, the new data will be appended to the end of the existing data file. Finally, select Text as the Email type and click OK. StatPac will advise you if any errors were encountered during the import. If so, the notepad will appear on the menu bar. Click Notepad on your menu bar to see a description of the errors.

### ***Setting Defaults for the Email Import***

An e-mail consist of two parts. The first part is the e-mail header and the second part is the contents (or body) of the e-mail. The header contains many lines that are often hidden by e-mail readers, but can be seen by loading an e-mail into the notepad. StatPac must be able to properly identify where the header starts and stops in order to know where the e-mail body begins. The settings in the StatPac.ini file may be adjusted to be compatible with your e-mail reader or language.

The StartEmailHeader and EndEmailHeader settings should be set to the text that begins and ends the header section. The StartEmailHeader parameter should be the text that begins the header section, and the EndEmailHeader parameter should be set to the last Email header line. If you are manually copying and pasting incoming e-mails to mailbox file, it may be important to change these settings. The default values for these parameters are:

StartEmailHeader = Return-Path:

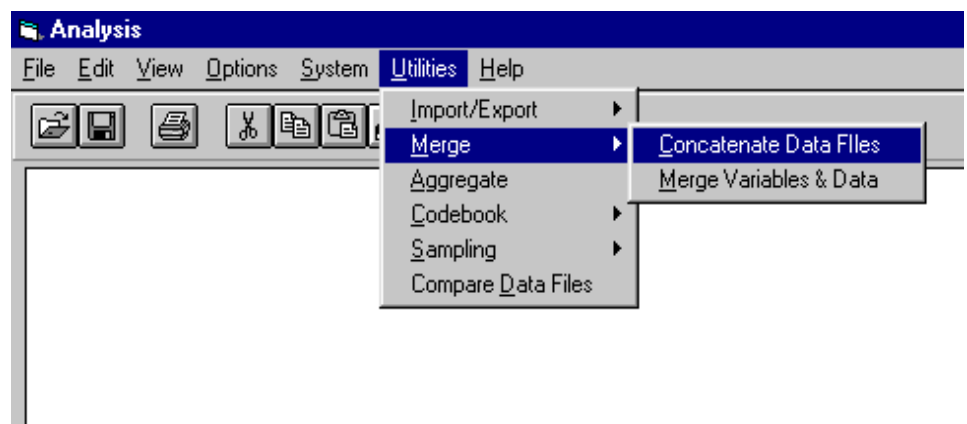
EndEmailHeader = X-UIDL:

Other e-mail parameters may also be set in the StatPac.ini file. StartEmailField and EndEmailField can be used to change the brackets from [ and ] to other characters. The EmailPrefix parameter tells StatPac what line contains the name/e-mail address of the respondent. The EmailVarName is the name of the StatPac codebook variable that will automatically capture the respondent's e-mail address in a plain-text e-mail, and the EmailDateField parameter is used to get the date of the e-mail in order to more precisely report which e-mails contained errors. By modifying these parameters, StatPac can be made to work with any e-mail reader or language.

---

## Merging Data Files

There are two basic ways that data files can be merged. The first is called concatenation, and it is used to merge two or more data files that contain the same variables in the same order. The second type of merge lets you join data containing different variables. Select Utilities, Merge, and then the type of merge you want to perform.



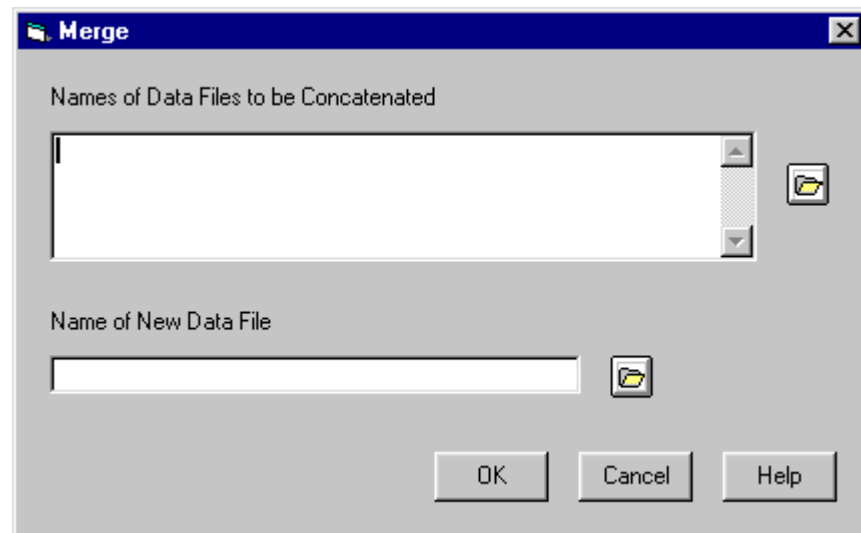
## Concatenate Data Files

Many times, several data entry operators will simultaneously enter data into a data file on their own machines. When all the data files have been entered, they can be merged into one large file by concatenating (joining) the data files.

For example, let's say you have three months of data in three separate files (JAN.DAT, FEB.DAT and MAR.DAT). The following DOS command would create a new file called QUARTER1.DAT which contained all three months of data. You could then run your analysis on all the data for the first quarter.

The concatenation-style merge assumes that the codebook(s) for all the data files are exactly the same. The Merge program will let you concatenate any number of data files into a new (larger) data file. You can type the data file names or use the browse button to select data files. Only one data file name should appear per line.

Do not confuse concatenating files with the MERGE utility program. If all your data files reference identical study information (contain the same variables in the same order), use concatenation to merge your data into one file. If your data files, however, contain different variables, use the MERGE utility program.



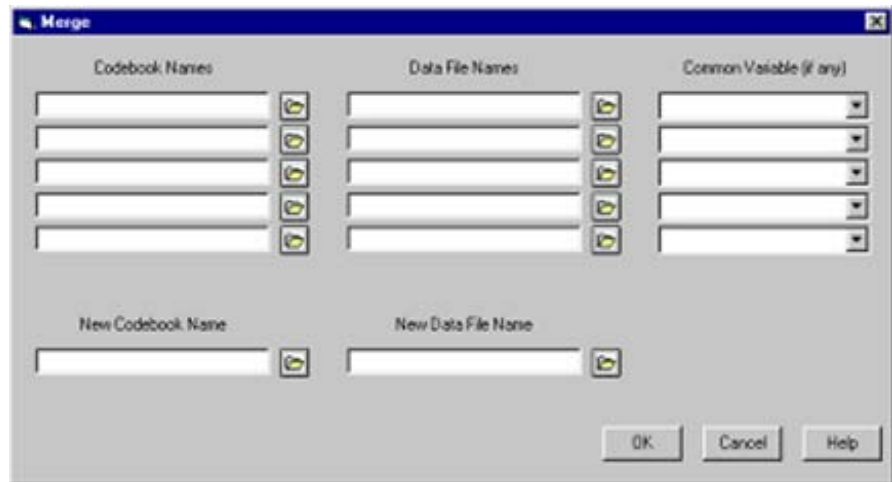
## Merge Variables and Data

The merge program allows you to extract selected variables from up to five studies and create an entirely new study that will be saved on disk. If data files have already been entered for any of the studies, they can also be restructured to match the new study format.

Do not confuse the function of this program with data file concatenation. If two data files have identical formats (i.e., they contain the same variables in the same order), the data files should be merged with the concatenation program.

The restructure and merge program can be used to reorganize a single study (and data file) or to combine several studies (and their associated data files). It allows complete versatility with regard to which variables are selected from each of the studies and the order of the variables.

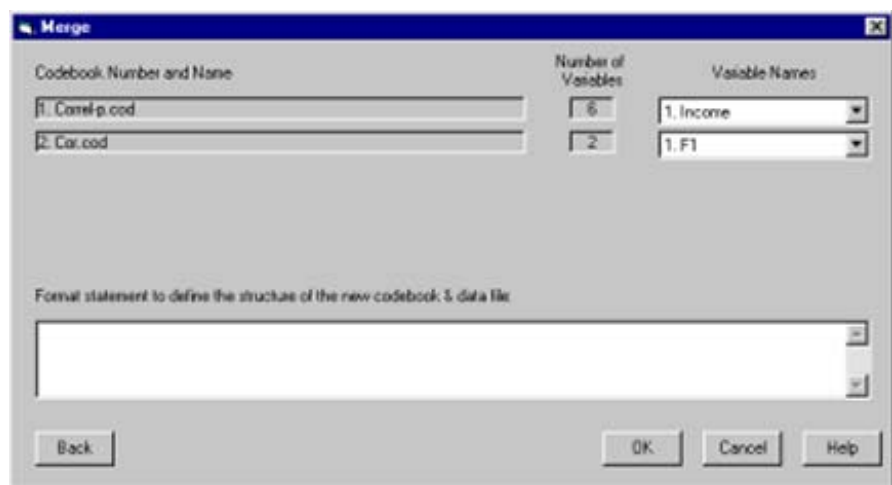
The program will ask for the name(s) of the codebooks, data files, and common variables that will be utilized. For each specified codebook, also enter the name of the associated data file (if one exists). If no data file is specified for a particular study, the program will use blanks for all variables requested from that study.



Also select the common variable in each of the studies. This refers to a variable that can be used to match up the records from each data file (e.g., "CASE ID"). If there is not a common variable, it is imperative that the data files contain the same number of records and in the same order. That is, record one from data file one should represent the same respondent (case) as record one from data file two.

If a data record is missing in any of the data files, it could cause data from one file to be matched with the wrong data from another file. Therefore, it is always a good idea to have a common variable in each of the data files (and associated study information) that represents a unique case identification number. **All data files must be sorted by this variable before running this program.** If one of the data files is missing a particular record, blanks will be merged into the output file.

Click OK to continue. The study numbers and names will be displayed, and the program will request the format for the new study. The format statement defines the structure of the new codebook.



The general format for creating a new file structure is:

(<Study number>) <Variables> or <Variable range>

An example of a format statement is:

(1) 1-3,8,4 (3) 2-7 (2) 9 14 (1) 12

This statement indicates that the new study format should contain variables in the following order:

From study 1 - variables 1, 2, 3, 8 and 4

From study 3 - variables 2, 3, 4, 5, 6 and 7

From study 2 - variables 9 and 14

From study 1 - variable 12

Notice that the study number is enclosed in parentheses with no spaces. Individual variables may be separated by either commas or spaces. A range of variables is specified by a dash (minus sign) with no spaces on either side of the dash. If the format statement requires more than one line, just continue typing and word-wrap will correctly break the line

Variables may be specified in any order. The study numbers will be displayed at the top of the screen and are assigned by the computer simply for convenience when specifying the new study format. The individual variable numbers for each codebook can be determined by examining the Variable Names windows

The new study format will be checked for validity before processing begins. If errors are found, you will be asked to re-enter the format. The new study and data file (if specified) will be written.

---

## Aggregate

The aggregate utility program creates a new study and data file that consist of aggregate statistics for subgroups of the original data. Any descriptive statistic may be included in the aggregate files. The program allows the creation of both compositional and true aggregate files.

For example, let's say we've distributed a questionnaire to 200 people in each of 50 communities. After performing some preliminary analyses, we want to compare the communities on a number of the interval or ratio-type questions. We could, of course, use the IF-THEN SELECT and WRITE commands to create subfiles for each of the communities and then perform descriptive statistics analyses on each of the subfiles. Obviously, this would be a very time consuming procedure. The aggregate utility program provides a much more efficient way to derive this information.

By using the aggregate program, we could create a new codebook and data file that just contain the descriptive statistics we desire. Each record in the new aggregate file would represent one community. The record would contain the descriptive statistics for the community as a whole (and not the raw data from the original file). Since there are 50 communities, the aggregate file would contain 50 records. This type of aggregate file is called a true aggregate file. It is made up of just the aggregate statistics and does not contain the original data collected. After creating a true

aggregate file, the LIST command could be used to print a summary of the descriptive statistic for the communities.

The other type of aggregate file is referred to as a compositional file. Using the same example as above, let's say we want to compare each case in our original file to the descriptive statistic for the community. For example, we might want to compare the individual's age with the mean age in that person's community. In other words, we want each record in the aggregate data file to contain both the original raw data and the descriptive statistic for the community as a whole. The number of records in the compositional aggregate file will contain the same number of records as the original raw data file. However, the aggregate file will contain more variables (the original variables plus the aggregate statistics).

When creating either a true or compositional aggregate file, a new study information file will also be automatically created to match the new aggregate data file.

**Before running the aggregate program, the data file must be sorted by the variable that contains the group code.** For example, if you plan to create an aggregate file by community, the data file must be sorted by community before running the aggregate utility program. The sort order is not important, however, it is important that all cases from the same community fall together in the file. The aggregate program will accommodate a minimum of 1000 individual groups.

To sort the file, you might use the following procedure:

```
STUDY GOVT
SORT (A) COMMUNITY
SAVE
..
```

Then run the Aggregate program. It will ask for the codebook name, data file and the variable containing the group code. This refers to the codebook and data file that already exist (not the new aggregate files). The variable containing the group code is the same variable that was used to sort the data file before running this program. In this example, it is the "community" variable. You must also select the type of aggregate file to be created, either compositional or true.

**Aggregate**

Codebook Name

Data File Name

Aggregate Type

☐ True

☒ Compositional

Grouping Variable

New Codebook Name

New Data File Name

OK Cancel Help

Click OK to continue. Now you can select the variable(s) for which you want to calculate aggregate statistics. Select the desired variable. Then click on the statistics you want for that variable. Each time you click on a statistic, an aggregate statement will be created in the Aggregate Statement window. Each aggregate statement will create one new aggregate variable.

**Aggregate**

Variables

6. Devote

6. Devote

7. Significant

8. Incentive

9. Clients

10. Firm

11. Opportunities

12. Win

13. Teamwork

1. Minimum

2. Maximum

3. Range

4. Sum

5. Mean

6. Median

7. Mode

8. Biased Variance

9. Biased SD

Variable No.	Statistic No.	Format for Statistic

OK Cancel Help



When performing a compositional aggregate procedure, the new aggregate variables will be added to the end of each data record. If the study and data file contain 10 variables, and you type two aggregate statements, the new aggregate variables would be added as variables 11 and 12.

When performing a true aggregate procedure, the first variable in the aggregate file will always be the group code (that is, the variable used to determine the groups). Each aggregate statement will produce a statistic that is added as the next variable in the file. The first aggregate statement would create variable two, the next variable three, and so forth.

Aggregate statistics can only be calculated for numeric-type variables. There is one exception to this rule: If the variable used to split the data file into groups is alpha, you may still calculate the number of valid cases. In our example, if community were coded alpha, it would be acceptable to ask for the number of valid cases (statistic 17) for this variable.

Each aggregate statement you enter will create a new variable in the aggregate file. After entering all the aggregate statements, click OK. A new codebook will be created. The new variable labels in this study will include both the original labels and the types of statistics. After the new study has been created, the program will perform all the aggregate calculations and write the new data file.

Because many calculations are involved in creating an aggregate file, the program may take some time to finish. It will display a message informing you of successful completion.

If any statistic cannot be calculated, or if there are an insufficient number of columns to hold the aggregate statistic, the output file will contain spaces for that variable. For example, if you requested the mode, and the group was multi-modal, the aggregate statistic would be stored as blanks.

---

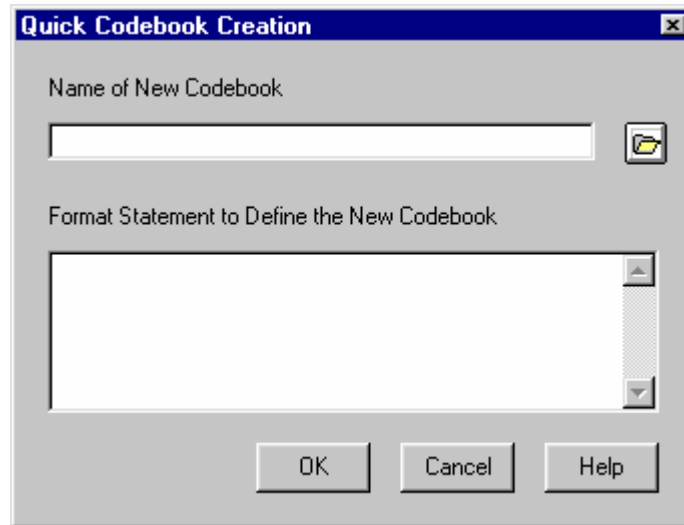
## Codebook

There are two utility programs for codebooks. The Quick Codebook Creation utility creates an entire codebook using a single FORTRAN-like statement. The Check Codebook & Data utility is used to verify the integrity of the codebook and to fix errors in the file.

### Quick Codebook Creation

The fastest way to create a codebook is to use the Quick Codebook Creation program. However, this will create a "barebones" codebook consisting of only the format for each variable. In most cases, you'll want to use the Grid or Variable Detail window to create a new codebook.

Select Analysis, Utilities, Codebook, Quick Codebook Creation. You will need to enter a file name for the new codebook and a *format statement*. This is essentially a data definition statement and is similar to a FORTRAN style format statement.



The Format Statement defines the number and type of variables that will be in the new study. **It is the combination of all the individual variable formats.** Using the format statement can save considerable time if variable and value labels are not required, or if you plan to use a fixed format data file from another source.

The syntax for each component of a format statement is:

<No. of Vars.> <Var. Type> <No. of Cols> . <Decimals>

<*No. of Vars.*> is the number of consecutive variables that use the format defined by the next three parameters. If this component of the format statement is omitted, the default is one.

<*Var. Type*> is always A or N and refers to whether the variable(s) are alpha or numeric. StatPac automatically left justifies alpha variables and right justifies numeric variables.

<*No. of Cols*> is the field width allocated for the variable(s). This is the total field width for the variable(s) and it must be large enough to hold a plus or minus sign and a decimal point if necessary.

. <*Decimals*> is the number of significant decimal places that the variable(s) will contain. This component of the format statement is optional and may be omitted. If <*decimals*> is not specified, the data will be stored exactly as entered (with or without a decimal point).

### ***Examples of Format Statements***

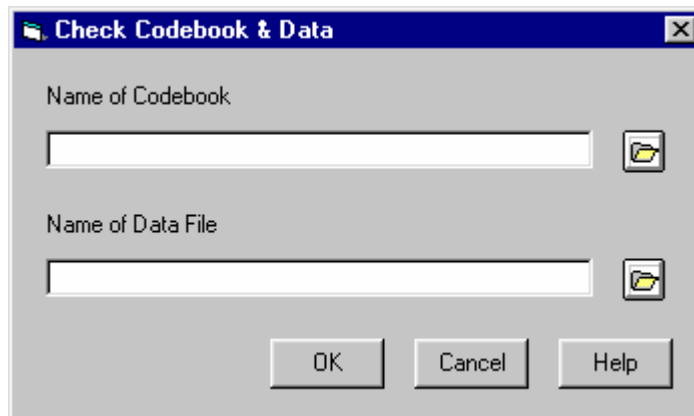
1N5	creates 1 numeric variable using 5 columns
N5	creates 1 numeric variable using 5 columns
12N3	creates 12 numeric variables each using 3 columns
N5.2	creates 1 numeric variable using 5 columns the format of the variable will be ###.##
7N2.0	creates 7 numeric variables each using 2 columns; the format of the variables will be ## (always rounded to an integer)

A1	creates 1 alpha variable using 1 column
2A35	creates 2 alpha variables each using 35 columns
5N4 2A1 3N7.2	creates a study with 10 variables. 1-5 are numeric each using 4 columns, 6-7 are alpha using 1 column each, 8-10 are numeric using 7 columns each with 2 significant decimal places

## Check Codebook and Data

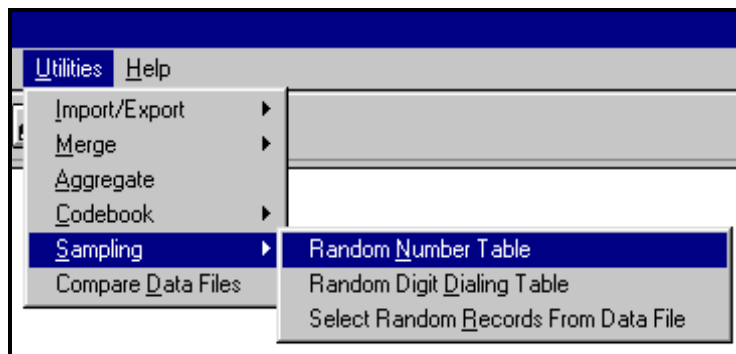
This utility program will verify the integrity of a codebook and data file. If errors are found the program will attempt to fix them. If you have created a codebook to match a foreign data file (one created by a program other than StatPac), use this program to make sure that the data record lengths match the codebook you created.

Select the codebook and data file to be checked and click OK. If the program corrects any errors, they will be listed in the notepad.



## Sampling

The Sampling program is used to generate a random number table, create a random digit dialing table for telephone studies, and to select a random sample from a data file.

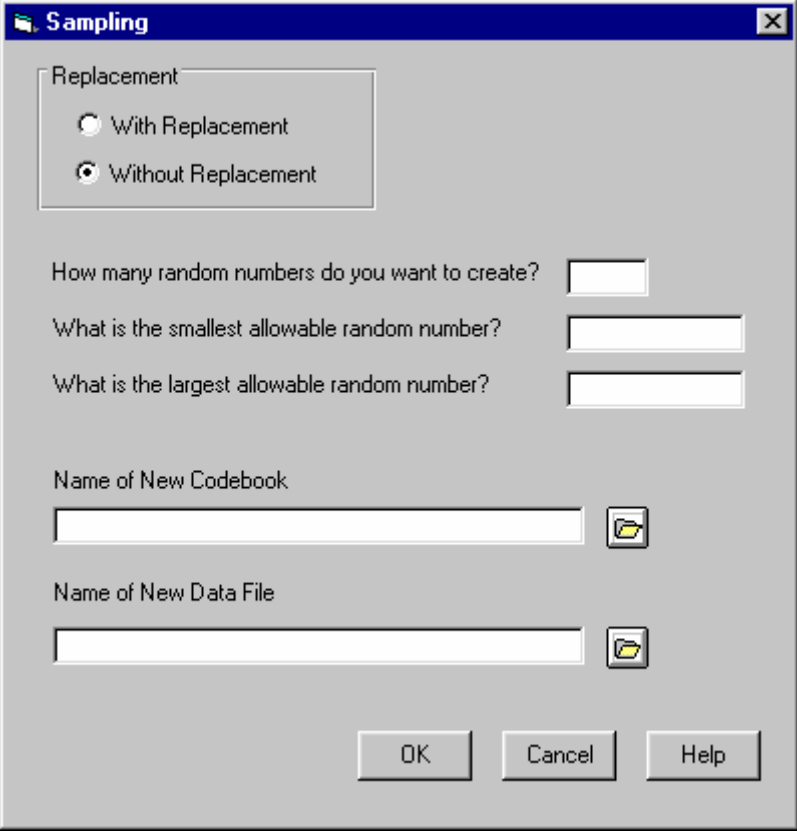


## Random Number Table

When planning to conduct a survey, choosing the sample is just as important as the survey itself. If the sample is incorrectly chosen, any results are likely to be distorted. That is, the characteristics of the sample will not represent the characteristics of the population.

One of the best ways to choose a sample is to use a random sampling technique. If the sample is randomly chosen from the population, it will represent the population. That is, characteristics of the sample are likely to be found in similar proportions in the population.

The classical method of selecting the sample is to give each case in the population a number and then randomly select numbers until the sample size is achieved. The second function of this program is to print a random number table.



You should first select whether the numbers should be selected with or without replacement. When replacement is used, a number may be selected more than once (selection does not eliminate it from being available for future selection). When random numbers are selected without replacement, the selection of a number eliminates it from the pool of available numbers. The algorithm used for selection without replacement will display the random numbers in sequential order.

Enter the number of random numbers you want to be printed. This relates to the sample size determined with the Statistics Calculator. Be sure to add a sufficient number to the ideal sample size to accommodate a pilot test and replacement of nonresponders (if part of your study design).

Enter the smallest allowable random number and the largest allowable random numbers. Typically, the lowest value would be one and the highest value would be equal to the number of cases in the population.

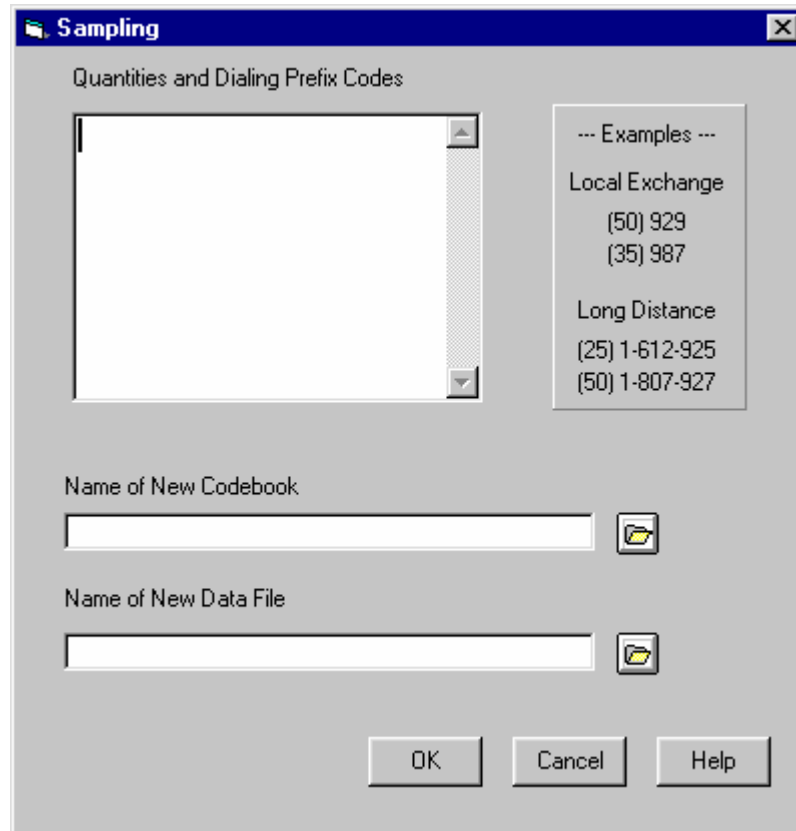
Enter the name of the StatPac codebook and data file to store the random numbers and click OK. A StatPac codebook and data file will be created that contains one variable called "RANDOM". Finally, the random numbers will be displayed in a compressed format in the Notepad. You do not need to save them with Notepad since they are already stored in a StatPac data file.

## Random Digit Dialing Table

Telephone surveys sometimes use random digit dialing to secure the sample. While this method will result in many non-working or non-voice numbers, it will produce a random sample of people who have telephones. Since local prefix codes are set (i.e., predefined by the phone company), only the last four digits of a phone number can be randomly selected. The random number method of creating a telephone file allows you to specify a series of local prefix codes and the number of random telephone numbers you want created for each prefix code.

There is an important consideration to keep in mind when creating a random digit file. Many of the random numbers will not be useful. For example, a number may be non-existent, a business office, or a fax or computer line. There are several algorithms for maximizing the number of home phone numbers, however, these techniques have generally produced poor results and are not included in StatPac. Therefore, it is usually a good idea to select more phone numbers than you actually need.

The random number utility program allows you to specify any number of prefixes and to specify how many numbers you want from each prefix. For local surveys, the prefix will be three digits (the local exchange); for long distance surveys, the prefix will be seven digits (i.e., 1 + three digits for the area code + three digits for the local exchange).



In the Local Exchange examples on the screen display, 50 numbers would be created with a 929 prefix and 35 numbers would be created with a 987 prefix. For the Long Distance examples, 25 numbers would be created that begin with 1-612-925 and 50 numbers would be created that begin with 1-807-927.

After you have finished typing the prefixes and quantities, click OK to create the phone number file. A StatPac codebook and data file will be created that contains one variable "TELEPHONE\_NUMBER". Finally, the random numbers will be displayed in a compressed format in the Notepad. You do not need to save them with Notepad since they are already stored in a StatPac data file.

The actual technique used to create the file is called random number selection without replacement. This means that as a phone number is selected, it will be eliminated from the pool of available numbers for the next selection. This eliminates the possibility of selecting the same number (with the same prefix) twice.

Depending on the number of prefixes and the quantities from each prefix, the actual creation of the file may take a little while. Please be patient; the program will inform you when the sample selection has been completed.

## Select Random Records from Data File

With this utility, you can select a specified number of random records from a data file and write them to a new data file. If you have a very large data base and a long procedure file, you might use this utility to create a shorter data file, and perform a test run of the procedure file on it.

Enter for the name of the existing data file, the new data file, and the number of records to be selected and written to the new data file.



You can also create a data file of the rejected (not selected) records. The file name for the rejected data records will be the same as the selected records file except that it will have a "-Rejected" suffix at the end of the file name.

To evoke this feature, you need to change a setting in the System Defaults File. Select File>Open>System Defaults File. Select Edit>Find and search for:  
CreateRejectedFile = 0

Change it to:

CreateRejectedFile = 1

Click the Save Icon (diskette). Select File>Close

The rejected records file will then be created along with the selected records file.

## Create Variable for Weighting

This utility will create a weighting variable that can be used with the WT option in order to adjust the data to conform to known population parameters. Up to ten variables may be weighted simultaneously. The mathematical technique is known by several names: *rim weighting*, *iterative proportional fitting*, *biproportional fitting*, *matrix raking*, and *matrix scaling*. All refer to the same general process of creating weights that adjust the data so selected demographic variables conform to actual known values in the population.

After conducting a survey, you might perform a frequency analysis of the gender variable and find that your sample isn't exactly the same as the gender distribution in the population. There may be numerous reasons for the difference, but any analyses you perform will reflect the sampling error.

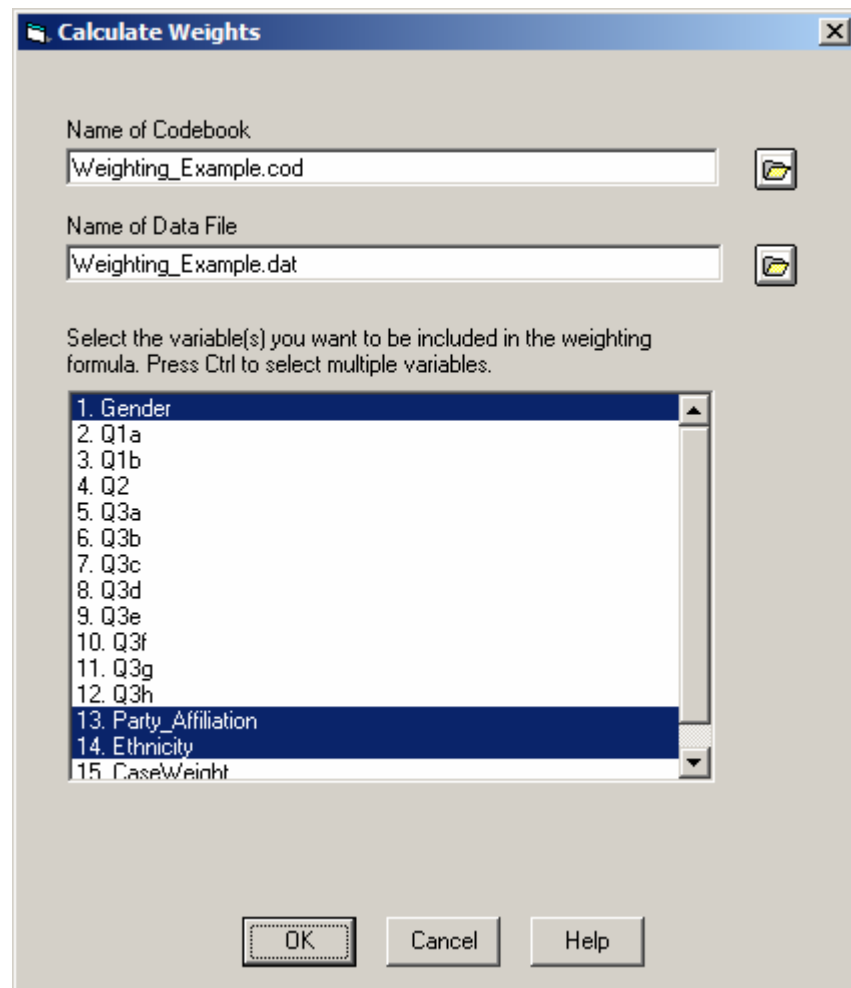
First try to understand the reason your sample is different from the population. The difference may simply be attributable to sampling error or it may be because of some hidden error in your sampling technique. After that, you can use this utility to create a set of weights that will compensate for the error in sampling. One or more variables may be adjusted.

This utility will create a new weighting variable and add it to the codebook and data file so that it can be referenced with the WT option in your procedures.

When there is only one demographic variable that needs to be adjusted, weights are easily calculated by hand as the desired percent divided by the observed percent. However, when more than one demographic variable needs adjustment (and they are done sequentially), subsequent weighting adjustments will modify previous adjustments and it's nearly impossible to make your sample match the population. Rim weighting lets you make all the adjustments simultaneously so the end result will be an adjusted sample that exactly reflects the known population parameters.

To use this utility, you only need to know the population parameters for the variables you will be weighting. That is, the percent in the population for each response category (value label).

When you start the utility program, you will be asked to select the codebook, data file, and the variables you will use for weighting. Use the browse button to select the codebook and the name of the data file will be filled in automatically. You can select one or more weighting variables. Hold the control key to make multiple selections if you want to weight by more than one variable.





After clicking OK, StatPac will read the data and calculate the observed frequency for each value label of the selected variable(s). These will be shown in parentheses adjacent to the value label. You can then enter the desired frequency (percent) for each of the value labels. The desired frequency is the population parameter. Enter a "DESIRED %" for each value label. After weighting, a frequency analysis with the WT option will show the desired percents instead of the observed percents and all other variables will be adjusted accordingly.

**Calculate Weights**

Generate CaseWeight Variable

New variable name for CaseWeight variable:  
CaseWeight

The following are the observed percents for each value label of the selected variable(s). Type the DESIRED PERCENT for each value label. Then click the Calculate button to generate the CaseWeight variable.

V1: Gender  
1=Male (69.2%) DESIRED % = 55  
2=Female (30.8%) DESIRED % = 45

V13: Party\_Affiliation  
3=Democrat (40.2%) DESIRED % = 44.5  
2=Republican (34.2%) DESIRED % = 39.3  
1=Independent (25.6%) DESIRED % = 16.2

V14: Ethnicity  
1=white (76.9%) DESIRED % = 76  
2=Non-white (23.1%) DESIRED % = 24

Calculate Cancel

This procedure will create a weighting variable and add it to the end of the codebook and data file. You can give the new weighting variable any name you want. The default name is *CaseWeight*. If the variable already exists in the codebook and data, it will be replaced with the new values calculated by this utility program. If it doesn't exist, it will become the last variable in the study.

For any given variable, the desired percents you enter must sum to exactly 100 percent. Missing data will be assigned a weight of zero. This means that if any of the weighting variables contain missing data, the weighted sample size will be reduced by the number of missing cases. Respondents often refuse to answer some demographic questions. If you know the incidence in the population that refuses, you can add a value label to the codebook <blank>=Refused for that variable. If you don't know, then you have to accept the reduced sample size.

When you click the Calculate button, StatPac will calculate values for the weighting variable and add them to the codebook and data file. To use the weighting variable, you can either add the WT=(CaseWeight) option to each procedure. If you're running a series of procedures as a batch, you can add WT=(CaseWeight)! (with the exclamation point) to just the first procedure being run and the WT option will be applied to all the procedures in the batch.

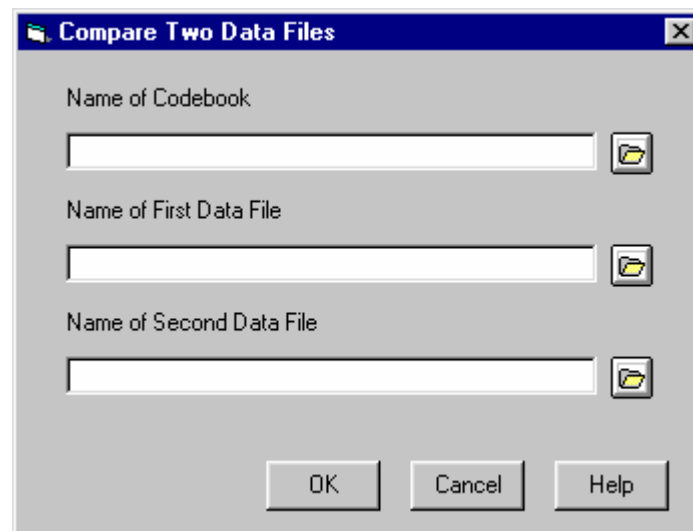
A word of caution. Weighting with the WT option counts respondents' answers based on their weight. Somebody with a weight of 2 will have twice the influence on the results as a person with a weight of 1. If your sample is markedly different from the population, weighting may introduce undesirable errors into the results by down-playing the influence of some respondents and up-playing others. Be careful.

---

## Compare Data Files

Many data entry operators use a double entry method of data verification. Data is entered into one data file and the same data is re-entered into another data file. The two data files are then compared for differences.

The purpose of this utility program is to identify possible errors in the data; it does not have any editing features.



Enter the name of the StatPac codebook and the names of the two data files to be compared. The data files should contain the same number of records in the same order.

Upon completion, the total number of errors will be reported. If differences are found, the record numbers and which variables are different will be shown in the Notepad. Use the notepad to print the errors listing

---

## Conversions

StatPac supports only two data types, alpha and numeric. This can make it difficult to work with dates and currency variables. These utilities simplify the task of working with date and currency variables.

The conversion utilities read an existing codebook and data file, and create a new codebook and data file with a new converted variable. The original date or currency variable is not modified and will remain “as is” in the codebook and data. Instead, a new variable (the converted field) is created and added to the end of the codebook and data.

## Date Conversions

The most common functions with dates are sorting and selecting. Typically, a user would create an alpha variable for a date variable because it contains non-numeric characters such as slashes or dashes. Regardless of the format, sorting by date or selecting the records between two dates can be difficult unless the date can be readily converted to a numeric eight-column (N8) variable in the format YYYYMMDD.

The screenshot shows a Windows-style dialog box titled "Date Conversions". It has a "Function" section with three radio buttons: "Create date sort key", "Calculate number of days between 2 dates", and "Expand date to text". Below this are three text input fields, each with a folder icon to its right: "Name of Codebook", "Name of Data File", and "Name of New Codebook and Data File". A label "Hold Control Key to Select Multiple Variables" is positioned above a large empty rectangular area. At the bottom are three buttons: "OK", "Cancel", and "Help".

The first function will take one or more date variables in any format and create new N8 variable(s) in YYYYMMDD format. The new N8 variable(s) can be used with the Sort command to sort a file by date. It can also be used with the Select command to select a range of dates.

The second function will calculate the number of days between two dates. The two dates can be any date format and the new variable (number of days) will be an N5 format. The absolute value of the difference between the two dates will be calculated and added to the end of the new codebook and data file.

The third function will create an English text version of a date in “D Mon, YYYY” format (e.g., 5 Oct 2005). The purpose is to make it possible for the user to subsequently use the List command to create an easily readable listing of the data.

## Currency Conversion

The currency conversion utility is useful for adding or removing the \$ or £ symbols, interpreting a K or M suffix, and removing commas from currency fields.

When conducting internet surveys (where the respondent is entering their own response) currency fields can create problems. You can require numeric input but that is often frustrating for respondents who want to enter something like 50K or 10M or \$25,000. If you believe respondents will want to enter anything other than a number, you can specify the field as alpha in the codebook (which will accept any input from the respondent). After the survey is closed, use this utility to convert the data to a numeric field.

The CurrencySymbol setting in the defaults (StatPac.ini) file can be set to your country’s currency symbol. When converting the alpha field to a number, commas will be removed, the letter K will multiple the value times a thousand, and the letter M will multiple the value times a million.

Conversion Type

☒ Convert Alpha Currency Field(s) To Numeric

☐ Convert Numeric Field(s) To Alpha Currency

☐ Round to whole dollar

Name of Codebook

Name of Data File

Name of New Codebook and Data File

Hold Control Key to Select Multiple Variables

OK Cancel Help

## Dichotomous Multiple Response Conversion

The dichotomous multiple response conversion utility is useful when you have imported data from an external source that coded multiple response variables in a dichotomous format.

For example, data in the external file might be coded as ones and blanks, where a one means the respondent selected the attribute and blank means they didn't.

Assume the question was *"What are your favorite colors?"* Imported data might look like this:

After importing, you could write a procedure to convert the data to the multiple response format used by StatPac. It would look like this:

```
Labels V1=What are your favorite colors?
Labels V2=What are your favorite colors?
Labels V3=What are your favorite colors?
Labels V4=What are your favorite colors?
Labels V1-V4 (1=Orange)(2=Blue)(3=Yellow)(4=Red)
Recode V2 (1=2)
Recode V3 (1=3)
Recode V4 (1=4)
Frequencies V1-V4
Options MR=Y
..
```

This will work fine although it is cumbersome. When there are ten or more variables in the multiple response group, it becomes more difficult because the imported variables are likely coded as N1, while the StatPac variables need to be coded as N2.

Two methods are incorporated into StatPac to deal with imported data that use dichotomous multiple response.

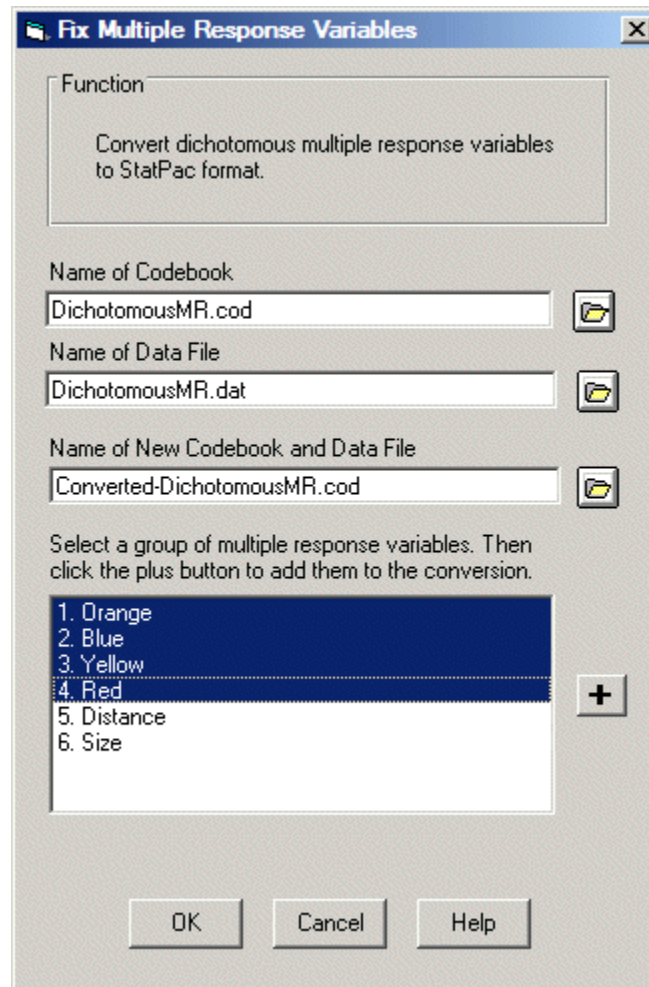
The first method is in the frequencies program itself. The MX=Code option can be added to the frequency program to tell StatPac that the variables are dichotomous. Then "Code" is the single character value that indicates the item is selected. In the above example, the data was coded as ones and blanks, so MX would be set to 1. If the data had been coded as Y and N, then MX would be set to Y.

```
Frequencies V1-V4
Options MR=Y MX=1
..
```

Using this method does not actually change the data file. StatPac just reads the data differently for the frequencies procedure. An exclamation mark cannot be used to permanently set the MX option. It must be explicitly specified in each procedure where you want to use it.

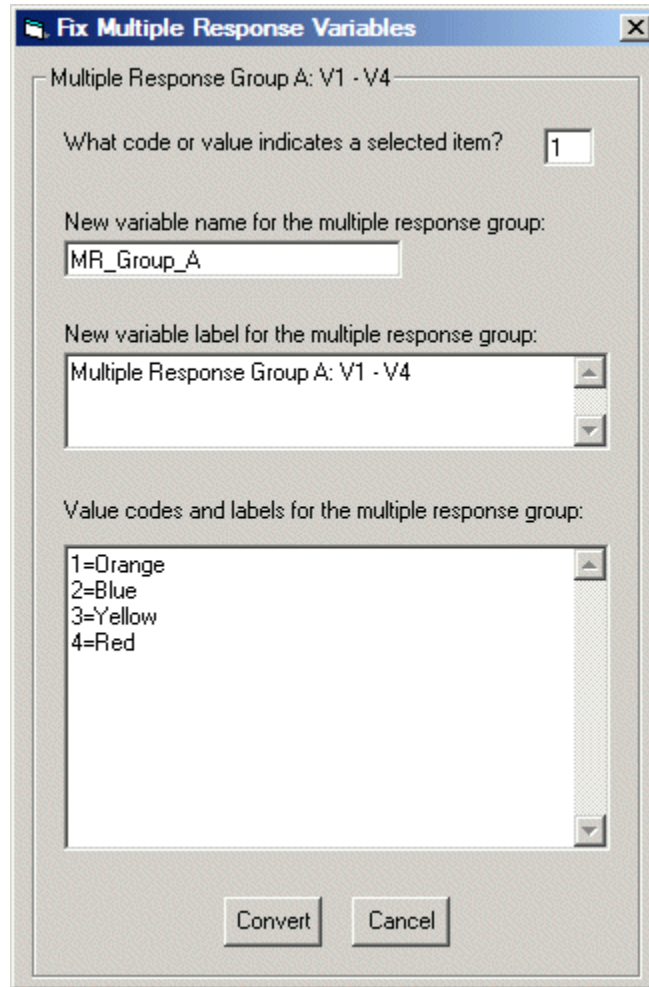
The other method is to actually convert the data file to the format used by StatPac for multiple response. If you plan to do banners or other procedures that utilize the dichotomous multiple response variables, then it is best to permanently alter the data. After conversion, the above data set would look like this:

The conversion utility lets you select several sets of variables that are dichotomous multiple response, however they are done one at a time. The first screen lets you select the codebook and data file, and specify a name for the new (converted) codebook and data file. After selecting the codebook, the variable names will appear so they can be selected.



After selecting the variables that make up the first multiple response group, click the plus button to add them to the conversion list.

The second screen lets you set the code and labeling for the selected group of variables.



The code is the dichotomous value that indicates the item is selected.

Since the imported data doesn't have a single variable name for the group of variables, StatPac names them MR\_Group\_A, MR\_Group\_B, etc. After the conversion, the converted variables in the group will be named using the \_x convention (e.g., MR\_Group\_A\_1, MR\_Group\_A\_2, MR\_Group\_A\_3, and MR\_Group\_A\_4). Thus, you might want to change the variable name to something more meaningful. For example, if you changed the name to Color, the converted variables would be named Color\_1, Color\_2, Color\_3, and Color\_4.

Similarly, the variable label might be changed to the actual question. All the converted variables will use that variable label. You could change "Multiple Response Group A: V1-V4" to "What are your favorite colors?"

After you are satisfied with the conversion labeling, click the Convert button. This will return you to the first screen where you can select an additional set of multiple response variables and click the plus button to add them to the conversion.

After you have finished selecting all the groups of multiple response variables, click OK to perform the conversion. The new codebook and new data file will then contain the multiple response variables in StatPac format.



# Statistics Calculator

---

## Statistics Calculator Menu

**Statistics Calculator** is an easy-to-use program designed to perform a series of basic statistical procedures related to distributions and probabilities. Most of the procedures are called *inferential* because data from a sample is used to infer to a population.

The menu bar of **Statistic Calculator** contains six types of operations that can be performed by the software.

Distributions	Counts	Percents
Means	Correlation	Sampling

The **Distributions** menu item is the electronic equivalent of probability tables. Algorithms are included for the z, t, F, and chi-square distributions. This selection may be used to find probabilities and critical values for the four statistics.

The **Counts** menu item contains routines to analyze a contingency table of counts, compute Fisher's exact probability for two-by-two tables, use the binomial distribution to predict the probability of a specified outcome, and the poisson distribution to test the likelihood of observing a specified number of events.

The **Percents** menu item is used to compare two percents. Algorithms are included to compare proportions drawn from one or two samples. There is also a menu option to calculate confidence intervals around a percent.

The **Means** menu item is used to calculate a mean and standard deviation of a sample, compare two means to each other, calculate a confidence interval around a mean, compare a sample mean to a population mean, compare two standard deviations to each other, and compare three or more standard deviations.

The **Correlation** menu item is used to calculate correlation and simple linear regression statistics for paired data. Algorithms are included for ordinal and interval data.

The **Sampling** menu item is used to determine the required sample size for a study. The software can be used for problems involving percents and means.

---

## Distributions Menu

The **Distributions** menu selection is used to calculate critical values and probabilities for various distributions. The most common distributions are the  $z$  (normal) distribution,  $t$  distribution,  $F$  distribution, and the chi-square distribution. Within the last 20 years, computers have made it easy to calculate exact probabilities for the various statistics. Prior to that, researchers made extensive use of books containing probability tables.

### Normal distribution

The normal distribution is the most well-known distribution and is often referred to as the  $z$  distribution or the bell shaped curve. It is used when the sample size is greater than 30. When the sample size is less than 30, the  $t$  distribution is used instead of the normal distribution.

The menu offers three choices: 1) probability of a  $z$  value, 2) critical  $z$  for a given probability, and 3) probability of a defined range.

#### ***Probability of a $z$ Value***

When you have a  $z$  (standardized) value for a variable, you can determine the probability of that value. The software is the electronic equivalent of a normal distribution probability table. When you enter a  $z$  value, the area under the normal curve will be calculated. The area not under the curve is referred to as the *rejection region*. It is also called a *two-tailed* probability because both tails of the distribution are excluded. The **Statistics Calculator** reports the two-tailed probability for the  $z$  value. A one-tailed probability is used when your research question is concerned with only half of the distribution. Its value is exactly half the two-tailed probability.

##### **Example**

$z$ -value: 1.96

-----

Two-tailed probability = .0500

#### ***Critical $z$ for a Given Probability***

This menu selection is used to determine the critical  $z$  value for a given probability.

##### **Example**

A large company designed a pre-employment survey to be administered to perspective employees. Baseline data was established by administering the survey to all current employees. They now want to use the instrument to identify job applicants who have very high or very low scores. Management has decided they want to identify people who score in the upper and lower 3% when compared to the norm. How many standard deviations away from the mean is required to define the upper and lower 3% of the scores?

The total area of rejection is 6%. This includes 3% who scored very high and 3% who scored very low. Thus, the two-tailed probability is .06. The  $z$  value required to reject 6% of the area under the curve is 1.881. Thus, new applicants who score higher or lower than 1.881 standard deviations away from the mean are the people to be identified.

Two tailed probability: .06

-----  
z-value = 1.881

### ***Probability of a Defined Range***

Knowing the mean and standard deviation of a sample allows you to establish the area under the curve for any given range. This menu selection will calculate the probability that the mean of a new sample would fall between two specified values (i.e., between the limits of a defined range).

#### **Example**

A manufacturer may find that the emission level from a device is 25.9 units with a standard deviation of 2.7. The law limits the maximum emission level to 28.0 units. The manufacturer may want to know what percent of the new devices coming off the assembly line will need to be rejected because they exceed the legal limit.

Sample mean = 25.9

Unbiased standard deviation = 2.7

Lower limit of the range = 0

Upper limit of the range = 28.0

-----  
Probability of a value falling within the range = .7817

Probability of a value falling outside the range = .2183

The area under the curve is the sum of the area defined by the lower limit plus the area defined by the upper limit.

The area under the normal curve is the probability that additional samples would fall between the lower and upper limits. In this case, the area above the upper limit is the rejection area (21.83% of the product would be rejected).

## **T distribution**

Mathematicians used to think that all distributions followed the bell shaped curve. In the early 1900's, an Irish chemist named Gosset, discovered that distributions were much flatter than the bell shaped curve when working with small sample sizes. In fact, the smaller the sample, the flatter the distribution. The *t* distribution is used instead of the normal distribution when the sample size is small. As the sample size approaches thirty, the *t* distribution approximates the normal distribution. Thus, the *t* distribution is generally used instead of the *z* distribution, because it is correct for both large and small sample sizes, where the *z* distribution is only correct for large samples.

The menu offers three choices: 1) probability of a *t* value, 2) critical *t* value for a given probability, and 3) probability of a defined range.

### ***Probability of a t Value***

If you have a  $t$  value and the degrees of freedom associated with the value, you can use this program to calculate the two-tailed probability of  $t$ . It is the equivalent of computerized table of  $t$  values.

#### **Example**

t-value: 2.228

df: 10

-----  
Two-tailed probability = .050

### ***Critical t Value for a Given Probability***

This program is the opposite of the previous program. It is used if you want to know what critical  $t$  value is required to achieve a given probability.

#### **Example**

Two-tailed probability: .050

Degrees of freedom: 10

-----  
t-value = 2.228

### ***Probability of a Defined Range***

Knowing the mean and standard deviation of a sample allows you to establish the area under the curve for any given range. You can use this program to calculate the probability that the mean of a new sample would fall between two values.

#### **Example**

A company did a survey of 20 people who used its product. The mean average age of the sample was 22.4 years and the unbiased standard deviation was 3.1 years. The company now wants to advertise in a magazine that has a primary readership of people who are between 18 and 24, so they need to know what percent of its potential customers are between 18 and 24 years of age?

Sample mean: 22.4

Unbiased standard deviation: 3.1

Sample size = 20

Lower limit of the range = 18

Upper limit of the range = 24

-----  
Probability of a value falling within the range = .608

Probability of a value falling outside the range = .392

Because of the small sample size, the  $t$  distribution is used instead of the  $z$  distribution. The area under the curve represents the proportion of customers in the population expected to be between 18 and 24 years of age. In this example, we

would predict that 60.8% of the its customers would be expected to be between 18 and 24 years of age, and 39.2% would be outside of the range. The company decided not to advertise.

## F distribution

The  $F$ -ratio is used to compare variances of two or more samples or populations. Since it is a ratio (i.e., a fraction), there are degrees of freedom for the numerator and denominator. This menu selection may be use to calculate the probability of an  $F$  - ratio or to determine the critical value of  $F$  for a given probability. These menu selections are the computer equivalent of an  $F$  table.

### ***Probability of a F-Ratio***

If you have a  $F$ -ratio and the degrees of freedom associated with the numerator and denominator, you can use this program to calculate the probability.

#### **Example**

F-ratio: 2.774

Numerator degrees of freedom: 20

Denominator degrees of freedom: 10

-----  
Two-tailed probability = .0500

### ***Critical F for a Given Probability***

If you know the critical alpha level and the degrees of freedom associated with the numerator and denominator, you can use this program to calculate the F-ratio.

#### **Example**

Two-tailed probability = .0500

Numerator degrees of freedom: 20

Denominator degrees of freedom: 10

-----  
F-ratio: 2.774

## Chi-square distribution

The chi-square statistic is used to compare the observed frequencies in a table to the expected frequencies. This menu selection may be use to calculate the probability of a chi-square statistic or to determine the critical value of chi-square for a given probability. This menu selection is the computer equivalent of an chi-square table.

### ***Probability of a Chi-Square Statistic***

If you have a chi-square value and the degrees of freedom associated with the value, you can use this program to calculate the probability of the chi-square statistic. It is the equivalent of computerized table of chi-square values.

#### **Example**

Chi-square value: 18.307

Degrees of freedom: 10

-----  
Probability = .050

### ***Critical Chi-Square for a Given Probability***

If you have the critical alpha level and the degrees of freedom, you can use this program to calculate the probability of the chi-square statistic. It is the equivalent of computerized table of chi-square values.

#### **Example**

Probability = .0500

Degrees of freedom: 10

-----  
Chi-square value: 18.307

---

## **Counts Menu**

The **Counts** menu selection has four tests that can be performed for simple frequency data. The chi-square test is used to analyze a contingency table consisting of rows and columns to determine if the observed cell frequencies differ significantly from the expected frequencies. Fisher's exact test is similar to the chi-square test except it is used only for tables with exactly two rows and two columns. The binomial test is used to calculate the probability of two mutually exclusive outcomes. The poisson distribution events test is used to describe the number of events that will occur in a specific period of time.

### **Chi-square test**

The chi-square is one of the most popular statistics because it is easy to calculate and interpret. There are two kinds of chi-square tests. The first is called a one-way analysis, and the second is called a two-way analysis. The purpose of both is to determine whether the observed frequencies (counts) markedly differ from the frequencies that we would expect by chance.

The observed cell frequencies are organized in rows and columns like a spreadsheet. This table of observed cell frequencies is called a *contingency table*, and the chi-square test is part of a *contingency table analysis*.

The chi-square statistic is the sum of the contributions from each of the individual cells. Every cell in a table contributes something to the overall chi-square statistic. If a given cell differs markedly from the expected frequency, then the contribution of that cell to the overall chi-square is large. If a cell is close to the expected frequency for that cell, then the contribution of that cell to the overall chi-square is low. A large chi-square statistic indicates that somewhere in the table, the observed frequencies differ markedly from the expected frequencies. It does not tell which cell (or cells) are causing the high chi-square...only that they are there. When a chi-square is high, you must visually examine the table to determine which cell(s) are responsible. When there are exactly two rows and two columns, the chi-square statistic becomes inaccurate, and Yate's correction for continuity is often applied.

If there is only one column or one row (a one-way chi-square test), the degrees of freedom is the number of cells minus one. For a two way chi-square, the degrees of freedom is the number of rows minus one times the number of columns minus one.

Using the chi-square statistic and its associated degrees of freedom, the software reports the probability that the differences between the observed and expected frequencies occurred by chance. Generally, a probability of .05 or less is considered to be a significant difference.

A standard spreadsheet interface is used to enter the counts for each cell. After you've finished entering the data, the program will print the chi-square, degrees of freedom and probability of chance.

Use caution when interpreting the chi-square statistic if any of the cell frequencies are less than five. Also, use caution when the total for all cells is less than 50.

### **Example**

A drug manufacturing company conducted a survey of customers. The research question is: Is there a significant relationship between packaging preference (size of the bottle purchased) and economic status? There were four packaging sizes: small, medium, large, and jumbo. Economic status was: lower, middle, and upper. The following data was collected.

	lower	middle	upper
small	24	22	18
medium	23	28	19
large	18	27	29
jumbo	16	21	33

-----  
Chi-square statistic = 9.743

Degrees of freedom = 6

Probability of chance = .1359

## **Fisher's Exact Test**

The chi-square statistic becomes inaccurate when used to analyze contingency tables that contain exactly two rows and two columns, and that contain less than 50 cases. Fisher's exact probability is not plagued by inaccuracies due to small N's. Therefore, it should be used for two-by-two contingency tables that contain fewer than 50 cases.

### **Example**

Here are the results of a recent public opinion poll broken down by gender. What is the exact probability that the difference between the observed and expected frequencies occurred by chance?

	Male	Female
Favor	30	42
Opposed	70	58

-----  
Fisher's exact probability = .0249

## Binomial Test

The binomial distribution is used for calculating the probability of dichotomous outcomes in which the two choices are mutually exclusive. The program requires that you enter the number of trials, probability of the desired outcome on each trial, and the number of times the desired outcome was observed.

### Example

If we were to flip a coin one hundred times, and it came up heads seventy times, what is the probability of this happening?

Number of trials: 100

Probability of success on each trial (0-1): .5

Number of successes: 70

-----  
Probability of 70 or more successes < .0001

## Poisson Distribution Events Test

The poisson distribution, like the binomial distribution, is used to determine the probability of an observed frequency. It is used to describe the number of events that will occur in a specific period of time or in a specific area or volume. You need to enter the observed and expected frequencies.

### Example

Previous research on a particular assembly line has shown that they have an average daily defect rate of 39 products. Thus, the expected number of defective products expected on any day is 39. The day after implementing a new quality control program, they found only 25 defects. What is the probability of seeing 25 or fewer defects on any day?

Observed frequency: 25

Expected frequency: 39

-----  
Probability of 25 or fewer events = .0226

---

## Percents Menu

Percents are understood by nearly everyone, and therefore, they are the most popular statistics cited in research. Researchers are often interested in comparing two percentages to determine whether there is a significant difference between them.

## Choosing the Proper Test

There are two kinds of t-tests between percents. Which test you use depends upon whether you're comparing percentages from one or two samples.



Every percentage can be expressed as a fraction. By looking at the denominator of the fraction we can determine whether to use a one-sample or two-sample t-test between percents. If the denominators used to calculate the two percentages represent the same people, we use a one-sample t-test between percents to compare the two percents. If the denominators represent different people, we use the two-sample t-test between percents.

For example suppose you did a survey of 200 people. Your survey asked,

*Were you satisfied with the program?*

☐ Yes   ☐ No   ☐ Don't know

Of the 200 people, 80 said yes, 100 said no, and 20 didn't know. You could summarize the responses as:

Yes                       $80/200 = .4 = 40\%$

No                         $100/200 = .5 = 50\%$

Don't know             $20/200 = .1 = 10\%$

Is there a significant difference between the percent saying yes (40%) and the percent saying no (50%)? Obviously, there is a difference; but how sure are we that the difference didn't just happen by chance? In other words, how reliable is the difference?

Notice that the denominator used to calculate the percent of yes responses (200) represents the same people as the denominator used to calculate the percent of no responses (200). Therefore, we use a one-sample t-test between proportions. The key is that the denominators represent the same people (not that they are the same number).

After you completed your survey, another group of researchers tried to replicate your study. They also used a sample size of 200, and asked the identical question. Of the 200 people in their survey, 60 said yes, 100 said no, and 40 didn't know. They summarized their results as:

Yes                       $60/200 = .3 = 30\%$

No                         $100/200 = .5 = 50\%$

Don't know             $40/200 = .2 = 20\%$

Is there a significant difference between the percent who said yes in your survey (40%) and the percent that said yes in their survey (30%)? For your survey the percent that said yes was calculated as  $80/200$ , and in their survey it was  $60/200$ . To compare the yes responses between the two surveys, we would use a two-sample t-test between percents. Even though both denominators were 200, they do not represent the same 200 people.

### **Examples that would use a one-sample t-test**

Which proposal would you vote for?

☐ Proposal A   ☐ Proposal B

Which product do you like better?

☐ Name Brand   ☐ Brand X

Which candidate would you vote for?

☐ Johnson   ☐ Smith   ☐ Anderson

When there are more than two choices, you can do the t-test between any two of them. In this example, there are three possible combinations: Johnson/Smith, Johnson/Anderson, and Smith/Anderson. Thus, you could actually perform three separate t-tests...one for each pair of candidates. If this was your analysis plan, you would also use Bonferroni's theorem to adjust the critical alpha level because the plan involved multiple tests of the same type and family.

### **Examples that would use a two-sample t-test**

A previous study found that 39% of the public believed in gun control. Your study found the 34% believed in gun control. Are the beliefs of your sample different than those of the previous study?

The results of a magazine readership study showed that 17% of the women and 11% of the men recalled seeing your ad in the last issue. Is there a significant difference between men and women?

In a brand awareness study, 25% of the respondents from the Western region had heard of your product. However, only 18% of the respondents from the Eastern region had heard of your product. Is there a significant difference in product awareness between the Eastern and Western regions?

## **One Sample t-Test between Percents**

This test can be performed to determine whether respondents are more likely to prefer one alternative or another.

### **Example**

The research question is: Is there a significant difference between the percent of people who say they would vote for candidate A and the percent of people who say they will vote for candidate B? The null hypothesis is: There is no significant difference between the percent of people who say they will vote for candidate A or candidate B. The results of the survey were:

Plan to vote for candidate A = 35.5%

Plan to vote for candidate B = 22.4%

Sample size = 107

The sum of the two percents does not have to be equal to 100 (there may be candidates C and D, and people that have no opinion). Use a one-sample t-test because both percentages came from a single sample.

Use a two-tailed probability because the null hypothesis does not state the direction of the difference. If the hypothesis is that one particular choice has a greater percentage, use a one-tailed test (divide the two-tailed probability by two).

Enter the first percent: 35.5  
Enter the second percent: 22.4  
Enter the sample size: 107

-----  
t-value = 1.808  
Degrees of freedom = 106  
Two-tailed probability = .074

You might make a statement in a report like this: A one-sample t-test between proportions was performed to determine whether there was a significant difference between the percent choosing candidate A and candidate B. The t-statistic was not significant at the .05 critical alpha level,  $t(106)=1.808$ ,  $p=.073$ . Therefore, we fail to reject the null hypothesis and conclude that the difference was not significant.

#### Two Sample t-Test between Percents

This test can be used to compare percentages drawn from two independent samples. It can also be used to compare two subgroups from a single sample.

#### Example

After conducting a survey of customers, you want to compare the attributes of men and women. Even though all respondents were part of the same survey, the men and women are treated as two samples. The percent of men with a particular attribute is calculated using the total number of men as the denominator for the fraction. And the percent of women with the attribute is calculate using the total number of women as the denominator. Since the denominators for the two fractions represent different people, a two-sample t-test between percents is appropriate.

The research question is: Is there a significant difference between the proportion of men having the attribute and the proportion of women having the attribute? The null hypothesis is: There is no significant difference between the proportion of men having the attribute and the proportion of women having the attribute. The results of the survey were:

86 men were surveyed and 22 of them (25.6%) had the attribute.

49 women were surveyed and 19 of them (38.8%) had the attribute.

Enter the first percent: 25.6  
Enter the sample size for the first percent: 86  
Enter the second percent: 38.8  
Enter the sample size for the second percent: 49

-----  
t-value = 1.603  
Degrees of freedom = 133  
Two-tailed probability = .111

You might make a statement in a report like this: A two-sample t-test between proportions was performed to determine whether there was a significant difference

between men and women with respect to the percent who had the attribute. The t-statistic was not significant at the .05 critical alpha level,  $t(133)=1.603$ ,  $p=.111$ . Therefore, we fail to reject the null hypothesis and conclude that the difference between men and women was not significant.

### **Another example**

Suppose interviews were conducted at two different shopping centers. This two sample t-test between percents could be used to determine if the responses from the two shopping centers were different.

The research question is: Is there a significant difference between shopping centers A and B with respect to the percent that say they would buy product X? The null hypothesis is: There is no significant difference between shopping centers A and B with respect to the percent of people that say they would buy product X. A two-tailed probability will be used because the hypothesis does not state the direction of the difference. The results of the survey were:

89 people were interviewed as shopping center A and 57 of them (64.0%) said they would buy product X.

92 people were interviewed as shopping center B and 51 of them (55.4%) said they would buy product X.

Enter the first percent: 64.0

Enter the sample size for the first percent: 89

Enter the second percent: 55.4

Enter the sample size for the second percent: 92

-----  
t-value = 1.179

Degrees of freedom = 179

Two-tailed probability = .240

You might write a paragraph in a report like this: A two-sample t-test between proportions was performed to determine whether there was a significant difference between the two shopping centers with respect to the percent who said they would buy product X. The t-statistic was not significant at the .05 critical alpha level,  $t(179)=1.179$ ,  $p=.240$ . Therefore, we fail to reject the null hypothesis and conclude that the difference in responses between the two shopping centers was not significant.

## **Confidence Intervals around a Percent**

Confidence intervals are used to determine how much latitude there is in the range of a percent if we were to take repeated samples from the population.

### **Example**

In a study of 150 customers, you find that 60 percent have a college degree. Your best estimate of the percent who have a college degree in the population of customers is also 60 percent. However, since it is just an estimate, we establish confidence intervals around the estimate as a way of showing how reliable the estimate is.

Confidence intervals can be established for any error rate you are willing to accept. If, for example, you choose the 95% confidence interval, you would expect that in five percent of the samples drawn from the population, the percent who had a college degree would fall outside of the interval.

What are the 95% confidence intervals around this percent? In the following example, note that no value is entered for the population size. When the population is very large compared to the sample size (as in most research), it is not necessary to enter a population size. If, however, the sample represents more than ten percent of the population, the formulas incorporate a finite population correction adjustment. Thus, you only need to enter the population size when the sample size exceeds ten percent of the population size.

Enter the percent: 60

Enter the sample size: 150

Enter the population size: (left blank)

Enter the desired confidence interval (%): 95

-----  
Standard error of the proportion = .040

Degrees of freedom = 149

95% confidence interval = 60.0%  $\pm$  7.9%

Confidence interval range = 52.1% to 67.9%

Therefore, our best estimate of the population proportion with 5% error is 60%  $\pm$  7.9%. Stated differently, if we predict that the proportion in the population who have a college degree is between 52.1% and 67.9%, our prediction would be wrong for 5% of the samples that we draw from the population.

---

## Means Menu

Researchers usually use the results from a sample to make inferential statements about the population. When the data is interval or ratio scaled, it is usually described in terms of central tendency and variability. Means and standard deviations are usually reported in all research.

### Mean and Standard Deviation of a Sample

This menu selection will let you enter data for a variable and calculate the mean, unbiased standard deviation, standard error of the mean, and median. Data is entered using a standard spreadsheet interface. Finite population correction is incorporated into the calculation of the standard error of the mean, so the population size should be specified whenever the sample size is greater than ten percent of the population size.

#### Example

A sample of ten was randomly chosen from a large population. The ten scores were:

20 22 54 32 41 43 47 51 45 35

-----  
Mean = 39.0

Unbiased standard deviation = 11.6

Standard error of the mean = 3.7

Median = 42.0

## Matched Pairs t-Test between Means

The matched pairs t-test is used in situations where two measurements are taken for each respondent. It is often used in experiments where there are before-treatment and after-treatment measurements. The t-test is used to determine if there is a reliable difference between the mean of the before-treatment and the mean of the after treatment measurements.

<u>Pretreatment</u>	<u>Posttreatment</u>
Johnny -----	Johnny
Martha -----	Martha
Jenny -----	Jenny

Sometimes, in very sophisticated (i.e., expensive) experiments, two groups of subjects are individually matched on one or more demographic characteristics. One group is exposed to a treatment (experimental group) and the other is not (control group).

<u>Experimental</u>	<u>Control</u>
Johnny -----	Fred
Martha -----	Sharon
Jenny -----	Linda

The t-test works with small or large N's because it automatically takes into account the number of cases in calculating the probability level. The magnitude of the t-statistic depends on the number of cases (subjects). The t-statistic in conjunction with the degrees of freedom are used to calculate the probability that the difference between the means happened by chance. If the probability is less than the critical alpha level, then we say that a significant difference exists between the two means.

### Example

A example of a matched-pairs t-test might look like this:

Pretest	Posttest
8	31
13	37
22	45
25	28

29	50
31	37
35	49
38	25
42	36
52	69

-----

Var.1: Mean = 29.5   Unbiased SD = 13.2

Var. 2: Mean = 40.7   Unbiased SD = 13.0

t-statistic = 2.69

Degrees of freedom = 9

Two-tailed probability = .025

You might make a statement in a report like this: The mean pretest score was 29.5 and the mean posttest score was 40.7. A matched-pairs t-test was performed to determine if the difference was significant. The t-statistic was significant at the .05 critical alpha level,  $t(9)=2.69$ ,  $p=.025$ . Therefore, we reject the null hypothesis and conclude that posttest scores were significantly higher than pretest scores.

## Independent Groups t-Test between Means

This menu selection is used to determine if there is a difference between two means taken from different samples. If you know the mean, standard deviation and size of both samples, this program may be used to determine if there is a reliable difference between the means.

One measurement is taken for each respondent. Two groups are formed by splitting the data based on some other variable. The groups may contain a different number of cases. There is not a one-to-one correspondence between the groups.

<u>Score</u>	<u>Sex</u>		<u>Males</u>	<u>Females</u>
25	M		25	27
27	F	-----becomes---->	19	17
17	F			21
19	M			
21	F			

Sometimes the two groups are formed because the data was collected from two different sources.

<u>School A Scores</u>	<u>School B Scores</u>
525	427
492	535

582  
554  
520

600

There are actually two different formulas to calculate the  $t$ -statistic for independent groups. The  $t$ -statistics calculated by both formulas will be similar but not identical. Which formula you choose depends on whether the variances of the two groups are equal or unequal. In actual practice, most researchers assume that the variances are unequal because it is the most conservative approach and is least likely to produce a Type I error. Thus, the formula used in **Statistics Calculator** assumes unequal variances.

### **Example**

Two new product formulas were developed and tested. A twenty-point scale was used to measure the level of product approval. Six subjects tested the first formula. They gave it a mean rating of 12.3 with a standard deviation of 1.4. Nine subjects tested the second formula, and they gave it a mean rating of 14.0 with a standard deviation of 1.7. The question we might ask is whether the observed difference between the two formulas is reliable.

Mean of the first group: 12.3

Unbiased standard deviation of the first group: 1.4

Sample size of the first group: 6

-----  
Mean of the second group: 14.0

Unbiased standard deviation of the second group: 1.7

Sample size of the second group: 9

-----  
t value = 2.03

Degrees of freedom = 13

Two-tailed probability = .064

You might make a statement in a report like this: An independent groups t-test was performed to compare the mean ratings between the two formulas. The t-statistic was not significant at the .05 critical alpha level,  $t(13)=2.03$ ,  $p=.064$ . Therefore, we fail to reject the null hypothesis and conclude that there was no significant difference between the ratings for the two formulas.

## **Confidence Interval around a Mean**

You can calculate confidence intervals around a mean if you know the sample size and standard deviation.

The standard error of the mean is estimated from the standard deviation and the sample size. It is used to establish the confidence interval (the range within which we would expect the mean to fall in repeated samples taken from the population). The



standard error of the mean is an estimate of the standard deviation of those repeated samples.

The formula for the standard error of the mean provides an accurate estimate when the sample size is very small compared to the size of the population. In marketing research, this is usually the case since the populations are quite large. Thus, in most situations the population size may be left blank because the population is very large compared to the sample. However, when the sample is more than ten percent of the population, the population size should be specified so that the finite population correction factor can be used to adjust the estimate of the standard error of the mean.

### **Example**

Suppose that an organization has 5,000 members. Prior to their membership renewal drive, 75 members were randomly selected and surveyed to find out their priorities for the coming year. The mean average age of the sample was 53.1 and the unbiased standard deviation was 4.2 years. What is the 90% confidence interval around the mean? Note that the population size can be left blank because the sample size of 75 is less than ten percent of the population size.

Mean: 53.1

Unbiased standard deviation: 4.2

Sample size: 75

Population size: (left blank -or- 5000)

Desired confidence interval (%): 90

-----  
Standard error of the mean = .485

Degrees of freedom = 74

90% confidence interval = 53.1  $\pm$  .8

Confidence interval range = 52.3 - 53.9

## **Compare a Sample Mean to a Population Mean**

Occasionally, the mean of the population is known (perhaps from a previous census). After drawing a sample from the population, it might be helpful to compare the mean of your sample to the mean of the population. If the means are not significantly different from each other, you could make a strong argument that your sample provides an adequate representation of the population. If, however, the mean of your sample is significantly different than the population, something may have gone wrong during the sampling process.

### **Example**

After selecting a random sample of 18 people from a very large population, you want to determine if the average age of the sample is representative of the average age of the population. From previous research, you know that the mean age of the population is 32.0. For your sample, the mean age was 28.0 and the unbiased standard deviation was 3.2. Is the mean age of your sample significantly different from the mean age in the population?

Sample mean = 28  
Unbiased standard deviation = 3.2  
Sample size = 18  
Population size = (left blank)  
Mean of the population = 32  
-----  
Standard error of the mean = .754  
t value = 5.303  
Degrees of freedom = 17  
Two-tailed probability = .0001

The two-tailed probability of the t-statistic is very small. Thus, we would conclude that the mean age of our sample is significantly less than the mean age of the population. This could be a serious problem because it suggests that some kind of age bias was inadvertently introduced into the sampling process. It would be prudent for the researcher to investigate the problem further.

## Compare Two Standard Deviations

The F-ratio is used to compare variances. In its simplest form, it is the variance of one group divided by the variance of another group. When used in this way, the larger variance (by convention) is the numerator and the smaller is the denominator. Since the groups might have a different sample sizes, the numerator and the denominator have their own degrees of freedom.

### Example

Two samples were taken from the population. One sample had 25 subjects and the standard deviation 4.5 on some key variable. The other sample had 12 subjects and had a standard deviation of 6.4 on the same key variable. Is there a significant difference between the variances of the two samples?

First standard deviation: 4.5  
First sample size: 25  
Second standard deviation: 6.4  
Second sample size: 12  
-----  
F-ratio = 2.023  
Degrees of freedom = 11 and 24  
Probability that the difference was due to chance = .072

## Compare Three or more Means

Analysis of variance (ANOVA) is used when testing for differences between three or more means.

In an ANOVA, the F-ratio is used to compare the variance between the groups to the variance within the groups. For example, suppose we have two groups of data. In the best of all possible worlds, all the people in group one would have very similar scores. That is, the group is cohesive, and there would be very little variability in scores within the group. All the people in group two would also have similar scores (although different than group one). Again, there is very little variability within the group. Both groups have very little variability within their group, however, there might be substantial variability between the groups. The ratio of the between groups variability (numerator) to the within groups variability (denominator) is the F-ratio. The larger the F-ratio, the more certain we are that there is a difference between the groups.

If the probability of the F-ratio is less than or equal to your critical alpha level, it means that there is a significant difference between at least two of groups. The F-ratio does not tell which group(s) are different from the others...just that there is a difference.

After finding a significant F-ratio, we do "post-hoc" (after the fact) tests on the factor to examine the differences between levels. There are a wide variety of post-hoc tests, but one of the most common is to do a series of special t-tests between all the combinations of levels for that factor. For the post-hoc tests, use the same critical alpha level that you used to test for the significance of the F-ratio.

### **Example**

A company has offices in four cities with sales representatives in each office. At each location, the average number of sales per salesperson was calculated. The company wants to know if there are significant differences between the four offices with respect to the average number of sales per sales representative.

Group	Mean	SD	N
1	3.29	1.38	7
2	4.90	1.45	10
3	7.50	1.38	6
4	6.00	1.60	8

Source	df	SS	MS	F	p
Factor	3	62.8	20.9	9.78	.0002
Error	27	57.8	2.13		
Total	30	120.6			

### Post-hoc t-tests

Group	Group	t-value	df	p
1	2	2.23	15	.0412
1	3	5.17	11	.0003
1	4	3.58	13	.0034
2	3	3.44	14	.0040

2	4	1.59	16	.1325
3	4	1.09	12	.0019

---

## Correlation Menu

Correlation is a measure of association between two variables. The variables are not designated as dependent or independent. The two most popular correlation coefficients are: Spearman's correlation coefficient rho and Pearson's product-moment correlation coefficient.

When calculating a correlation coefficient for ordinal data, select Spearman's technique. For interval or ratio-type data, use Pearson's technique.

The value of a correlation coefficient can vary from minus one to plus one. A minus one indicates a perfect negative correlation, while a plus one indicates a perfect positive correlation. A correlation of zero means there is no relationship between the two variables. When there is a negative correlation between two variables, as the value of one variable increases, the value of the other variable decreases, and vice versa. In other words, for a negative correlation, the variables work opposite each other. When there is a positive correlation between two variables, as the value of one variable increases, the value of the other variable also increases. The variables move together.

The standard error of a correlation coefficient is used to determine the confidence intervals around a true correlation of zero. If your correlation coefficient falls outside of this range, then it is significantly different than zero. The standard error can be calculated for interval or ratio-type data (i.e., only for Pearson's product-moment correlation).

The significance (probability) of the correlation coefficient is determined from the t-statistic. The probability of the t-statistic indicates whether the observed correlation coefficient occurred by chance if the true correlation is zero. In other words, it asks if the correlation is significantly different than zero. When the t-statistic is calculated for Spearman's rank-difference correlation coefficient, there must be at least 30 cases before the t-distribution can be used to determine the probability. If there are fewer than 30 cases, you must refer to a special table to find the probability of the correlation coefficient.

### Example

A company wanted to know if there is a significant relationship between the total number of salespeople and the total number of sales. They collect data for five months.

Var. 1	Var. 2
207	6907
180	5991
220	6810
205	6553
190	6190

-----

Correlation coefficient = .921

Standard error of the coefficient = .068

t-test for the significance of the coefficient = 4.100

Degrees of freedom = 3

Two-tailed probability = .0263

### **Another Example**

Respondents to a survey were asked to judge the quality of a product on a four-point Likert scale (excellent, good, fair, poor). They were also asked to judge the reputation of the company that made the product on a three-point scale (good, fair, poor). Is there a significant relationship between respondents perceptions of the company and their perceptions of quality of the product?

Since both variables are ordinal, Spearman's method is chosen. The first variable is the rating for the quality the product. Responses are coded as 4=excellent, 3=good, 2=fair, and 1=poor. The second variable is the perceived reputation of the company and is coded 3=good, 2=fair, and 1=poor.

Var. 1	Var. 2
4	3
2	2
1	2
3	3
4	3
1	1
2	1

-----  
Correlation coefficient rho = .830

t-test for the significance of the coefficient = 3.332

Number of data pairs = 7

Probability must be determined from a table because of the small sample size.

## **Regression**

Simple regression is used to examine the relationship between one dependent and one independent variable. After performing an analysis, the regression statistics can be used to predict the dependent variable when the independent variable is known. Regression goes beyond correlation by adding prediction capabilities.

People use regression on an intuitive level every day. In business, a well-dressed man is thought to be financially successful. A mother knows that more sugar in her children's diet results in higher energy levels. The ease of waking up in the morning often depends on how late you went to bed the night before. Quantitative regression adds precision by developing a mathematical formula that can be used for predictive purposes.

For example, a medical researcher might want to use body weight (independent variable) to predict the most appropriate dose for a new drug (dependent variable). The purpose of running the regression is to find a formula that fits the relationship between the two variables. Then you can use that formula to predict values for the dependent variable when only the independent variable is known. A doctor could prescribe the proper dose based on a person's body weight.

The regression line (known as the *least squares line*) is a plot of the expected value of the dependent variable for all values of the independent variable. Technically, it is the line that "minimizes the squared residuals". The regression line is the one that best fits the data on a scatterplot.

Using the regression equation, the dependent variable may be predicted from the independent variable. The slope of the regression line ( $b$ ) is defined as the rise divided by the run. The  $y$  intercept ( $a$ ) is the point on the  $y$  axis where the regression line would intercept the  $y$  axis. The slope and  $y$  intercept are incorporated into the regression equation. The intercept is usually called the constant, and the slope is referred to as the coefficient. Since the regression model is usually not a perfect predictor, there is also an error term in the equation.

In the regression equation,  $y$  is always the dependent variable and  $x$  is always the independent variable. Here are three equivalent ways to mathematically describe a linear regression model.

$$y = \text{intercept} + (\text{slope} \cdot x) + \text{error}$$

$$y = \text{constant} + (\text{coefficient} \cdot x) + \text{error}$$

$$y = a + bx + e$$

The significance of the slope of the regression line is determined from the  $t$ -statistic. It is the probability that the observed correlation coefficient occurred by chance if the true correlation is zero. Some researchers prefer to report the  $F$ -ratio instead of the  $t$ -statistic. The  $F$ -ratio is equal to the  $t$ -statistic squared.

The  $t$ -statistic for the significance of the slope is essentially a test to determine if the regression model (equation) is usable. If the slope is significantly different than zero, then we can use the regression model to predict the dependent variable for any value of the independent variable.

On the other hand, take an example where the slope is zero. It has no prediction ability because for every value of the independent variable, the prediction for the dependent variable would be the same. Knowing the value of the independent variable would not improve our ability to predict the dependent variable. Thus, if the slope is not significantly different than zero, don't use the model to make predictions.

The coefficient of determination ( $r$ -squared) is the square of the correlation coefficient. Its value may vary from zero to one. It has the advantage over the correlation coefficient in that it may be interpreted directly as the proportion of variance in the dependent variable that can be accounted for by the regression equation. For example, an  $r$ -squared value of .49 means that 49% of the variance in the dependent variable can be explained by the regression equation. The other 51% is unexplained.

The standard error of the estimate for regression measures the amount of variability in the points around the regression line. It is the standard deviation of the data points as they are distributed around the regression line. The standard error of the estimate can be used to develop confidence intervals around a prediction.

### **Example**

A company wants to know if there is a significant relationship between its advertising expenditures and its sales volume. The independent variable is advertising budget and the dependent variable is sales volume. A lag time of one month will be used because sales are expected to lag behind actual advertising expenditures. Data was collected for a six month period. All figures are in thousands of dollars. Is there a significant relationship between advertising budget and sales volume?

IV	DV
4.2	27.1
6.1	30.4
3.9	25.0
5.7	29.7
7.3	40.1
5.9	28.8

-----  
Model:  $y = 10.079 + (3.700 \cdot x) + \text{error}$

Standard error of the estimate = 2.568

t-test for the significance of the slope = 4.095

Degrees of freedom = 4

Two-tailed probability = .0149

r-squared = .807

You might make a statement in a report like this: A simple linear regression was performed on six months of data to determine if there was a significant relationship between advertising expenditures and sales volume. The t-statistic for the slope was significant at the .05 critical alpha level,  $t(4)=4.10$ ,  $p=.015$ . Thus, we reject the null hypothesis and conclude that there was a positive significant relationship between advertising expenditures and sales volume. Furthermore, 80.7% of the variability in sales volume could be explained by advertising expenditures.

---

## Sampling Menu

The formula to determine sample size depends upon whether the intended comparisons involve means or percents.

### Sample Size for Percents

This menu selection is used to determine the required size of a sample for research questions involving percents.

Four questions must be answered to determine the sample size:

1. Best estimate of the population size: You do not need to know the exact size of the population. Simply make your best estimate. An inaccurate population size will not seriously affect the formula computations. If the population is very large, this item may be left blank.
2. Best estimate of the rate in the population (%): Make your best estimate of what the actual percent of the survey characteristic is. This is based on the null hypothesis. For example, if the null hypothesis is "blondes don't have more fun", then what is your best estimate of the percent of blondes that do have more fun? If you simply do not know, then enter 50 (for fifty percent).
3. Maximum acceptable difference (%): This is the maximum percent difference that you are willing to accept between the true population rate and the sample rate. Typically, in social science research, you would be willing to accept a difference of 5 percent. That is, if your survey finds that 25 percent of the sample has a certain characteristic, the actual rate in the population may be between 20 and 30 percent.
4. Desired confidence level (%): How confident must you be that the true population rate falls within the acceptable difference (specified in the previous question)? This is the same as the confidence that you want to have in your findings. If you want 95 percent confidence (typical for social science research), you should enter 95. This means that if you took a hundred samples from the population, five of those samples would have a rate that exceeded the difference you specified in the previous question.

### **Example**

A publishing wants to know what percent of the population might be interested in a new magazine on making the most of your retirement. Secondary data (that is several years old) indicates that 22% of the population is retired. They are willing to accept an error rate of 5% and they want to be 95% certain that their finding does not differ from the true rate by more than 5%. What is the required sample size?

Best estimate of the population size: (left blank)

Best estimate of the rate in the population (%): 22

Maximum acceptable difference (%): 5

Desired confidence level (%): 95

-----  
Required sample size = 263

## **Sample Size for Means**

This menu selection is used to determine the required size of a sample for research questions involving means.

Three questions must be answered to determine the sample size:

1. Standard deviation of the population: It is rare that a researcher knows the exact standard deviation of the population. Typically, the standard deviation of the population is estimated a) from the results of a previous survey, b) from a pilot study, c) from secondary data, or d) or the judgment of the researcher.



2. Maximum acceptable difference: This is the maximum amount of error that you are willing to accept. That is, it is the maximum difference that the sample mean can deviate from the true population mean before you call the difference significant.

3. Desired confidence level (%): The confidence level is your level of certainty that the sample mean does not differ from the true population mean by more than the maximum acceptable difference. Typically, social science research uses a 95% confidence level.

### **Example**

A fast food company wants to determine the average number of times that fast food users visit fast food restaurants per week. They have decided that their estimate needs to be accurate within plus or minus one-tenth of a visit, and they want to be 95% sure that their estimate does differ from true number of visits by more than one-tenth of a visit. Previous research has shown that the standard deviation is .7 visits. What is the required sample size?

Population standard deviation: .7

Maximum acceptable difference: .1

Desired confidence interval (%): 95

-----  
Required sample size = 188

W

# Index

## 8

80-column format 415

## A

add to an existing data file 62  
agglomerative hierarchical cluster analysis 356  
aggregate 121–22, 121–22, 121–22, 232, 348, 411,  
420–23, 420–23, 420–23, 420–23, 420–23, 420–  
23, 420–23, 420–23  
analysis commands 176, 206, 247  
analysis of variance 373  
Analysis of variance 373, 375, 380–83, 380–83, 373,  
380–82, 380–82, 380–82, 380–82, 383–91, 383–  
91, 383–91, 384–86, 380–83, 384–91, 384–91,  
387–91, 409, 456  
AND operator 212  
ANOVA 373  
ANOVA table 379  
ASCII file 68, 414  
Auto Advance 35  
autocorrelation 323, 334  
Automatic form creation 50, 52  
Automatic Mean Row 297  
Automatic Page Title 295  
Automatic Record Advance 56–57, 63, 67  
Automatic Total Row 291, 297  
Automatic variable creation 52  
AVERAGE command 224–25, 224–25, 224–25, 224–  
25, 224–25, 224–25, 224–25

## B

Backing up 6  
backup 4, 6, 194–95, 194–95, 194–95, 194–95, 194–95,  
194–95, 194–95, 199, 209  
banners 174, 203–4, 203, 206, 242–45, 245–47, 245–  
47, 247, 281–92, 281–97, 283–87, 283–87, 284–  
86, 284–86, 284–86, 289–97, 289–97, 281–83,  
289–93, 289–93, 247, 294–97, 281–83, 281–83,

400–402, 400–402, 400–402, 402–3, 402–3, 402–  
3, 403, 436

BANNERS Command 245, 281, 285–86, 285–86, 285–  
86, 285–86, 285–86, 285–86, 285, 289, 290, 296  
batch 1, 7, 11, 179, 185–86, 185–86, 185–86, 185–86,  
189–90, 189–90, 189–90, 189–90, 241–42, 241–  
42, 189, 185–86, 241, 432  
beta weights 327  
Binomial Test 444–46, 444, 446  
Box-Cox transformations 319  
branching 9–10, 9–10, 9–10, 9–10, 9–10, 9–10, 34, 55,  
77–78, 77–118, 77–118, 77–118, 77–118, 85, 102,  
118, 126–27, 126–27  
BREAKDOWN Command 305

## C

canonical correlation 393  
canonical correlation analysis 365  
canonical redundancy analysis 397  
canonical variable 369  
Caps Only 35  
card-image 415–16, 415–16, 415  
Carroll-Green-Shaffer scaling 406  
Category Creation 260, 276, 284, 308, 371, 381  
central tendency 27, 299, 301, 451  
cgi mail 2, 77  
change in probability 339  
change the value labels 204–6, 204–6, 204–6, 206, 216  
Check Codebook and Data 425  
chi-square 25, 256, 275–79, 275–79, 275–79, 275–79,  
277–79, 277–79, 277–79, 294–95, 294–95, 294–  
95, 294–95, 294–95, 339, 382, 399, 401, 404,  
406–7, 406–7, 406–7, 406–7, 406–7, 406, 439–40,  
439–40, 439–40, 439–40, 439–40, 439–40, 439–  
40, 443–44, 443–45, 443–45, 443–45, 443–45,  
443–45  
chi-square statistic 339  
classification matrix 367  
classification table 340  
cluster analysis 355  
cluster centroids 360  
cluster membership 364  
clustering algorithms 357  
codebook 1, 6–8, 6–8, 6–9, 6–12, 6–9, 6–8, 9–12, 9–12,  
9–12, 10, 9–12, 29–30, 29–30, 29–30, 30, 35–41,  
35–42, 29, 36–42, 36–41, 36–41, 42, 45–52, 45–  
46, 45–51, 45–52, 35–41, 35–42, 49–52, 49–52,  
53, 59, 62, 69–71, 69, 73–76, 73–76, 73–118, 73–  
118, 73–118, 77–78, 84–85, 84–85, 88–89, 91,  
101, 118–20, 118–20, 120, 126, 129, 131, 134–41,  
134–41, 134–49, 88, 84–85, 143–46, 143–46, 147,  
149, 163, 166, 173–74, 173–74, 173–74, 173–74,  
173–74, 173, 177, 184, 194–202, 194–202, 194–  
202, 143–46, 194–202, 120, 209, 214, 215–16,  
215, 250, 284–86, 284–86, 284–86, 284, 286, 292,

411–15, 411–15, 411–20, 411–14, 411–14, 411–13, 416–21, 416–21, 416–25, 416–25, 173–74, 423, 425, 427–28, 427–28, 427–28, 428–30, 428–30, 428–34, 431–33, 431–33, 432–33, 411–20, 432, 436–38, 436, 438

Codebook Libraries 37

codebook name 84, 174, 195, 421

coding session 267, 269

coefficient of determination 321

coefficient of multiple correlation 327

coefficient of multiple determination 326

Cohen's Kappa 278

collinearity diagnostics 346

column marginal disparities 280

Comma Delimited 10, 12, 162, 411, 414–15, 414–15

Comment Line 175, 242

comments 142, 190, 211, 229, 242, 268, 271–73, 271, 273, 415

communalities 350

Compact Data File 66–67, 66–67, 66–67, 67

Compare Data Files 412, 432

complex OR statements 212

compositional aggregate 411, 421–23, 421–23, 421, 423

Compression 204, 287

COMPUTE command 193, 220, 222, 224, 227–28, 227–28, 227–28, 227–28, 227–28, 236

concatenation 417–18, 417–18, 417–18, 417–18, 417–18

confidence intervals 323, 332

Confidence intervals 28, 256, 303, 321–24, 321–24, 321–23, 321–24, 324, 327, 332, 335, 439, 450, 454, 458, 460

Contingency Coefficient 278, 401, 407

contingency table 278, 401, 404–7, 404–7, 404–7, 404–7, 439, 444

Continuation Line 175, 207, 213, 242, 356

continuity 444

convergence tolerance 340, 354

CORRELATE Command 313

correlation matrix 313–16, 313–16, 313–15, 313–15, 313–16, 316, 328–29, 328–29, 328–29, 328–29, 338, 341, 343–44, 343–45, 344, 345–47, 345–47, 345–47, 347, 350, 351–53, 343–44, 353, 366, 394–95, 394–95, 394–95, 394–95

correspondence analysis 400

COUNT command 224–25, 224–25, 224–25, 225

covariance matrix 343

Cramer's V 277

create a form 41, 50–51, 50, 51, 53

create a new variable 214–16, 214–16, 214–16, 214–16, 215–16, 222, 225, 230, 236, 285, 402, 423

Create New Data File 62

criterion function 361

Cronbach's alpha 19, 315–16, 315–16, 315–16, 316

cross-factor rotation 352

CROSSTABS Command 274

cumulative percents 255–56, 255–56, 255–56

curve fitting 317–18, 317–18, 317–18, 317–18, 317, 318

## D

DATA command 174, 197, 200

data entry control parameters 30, 35, 66

data entry form 6, 9, 29, 32, 45, 55, 195, 201, 215–16, 215–16

data file 1, 6–8, 6–8, 6–8, 6–8, 6–8, 10, 12, 30, 35, 39, 45–46, 45–46, 45–47, 45–47, 45–50, 49, 61–64, 61–64, 61–68, 61–62, 61–62, 64, 66–69, 66–72, 66–69, 67–69, 72, 73–75, 73–75, 73, 75, 85, 150, 153, 164–66, 164–66, 164–65, 164–65, 166, 173–74, 173–74, 173–74, 173–74, 173, 191, 194–202, 194–202, 194–202, 194–201, 208–11, 208–11, 208–11, 211, 215, 218–19, 218–19, 218–19, 218–19, 226, 232–34, 232–35, 232–35, 232–35, 232–35, 240, 244, 248, 255, 260, 267, 277, 285, 289–90, 289–90, 289–90, 289–90, 289, 305–8, 305, 307–9, 307–9, 307–10, 307–10, 309–10, 310, 315, 325, 335, 341, 349, 355, 357, 374–78, 374–78, 374–77, 374–77, 411–25, 411–21, 411–15, 411–33, 411–28, 418–27, 423–27, 427–33, 428–30, 374–78, 431–33, 436–38, 436, 438

Data File Format 68, 375–78, 375–78, 375–77, 375–77

data file name 62, 173, 197, 412–13, 412–13, 412–13, 412–13, 412, 418

data input fields 49, 51–52, 51–52, 51, 52, 57

Data Input Settings 56–57, 56–57, 56–57, 56–57, 63, 66–67, 66–67, 66–67, 66–67, 67

data manager 1, 9–10, 9–10, 9–10, 9–10, 49, 51, 61–62, 61–62, 61–62, 61–62, 61, 62, 66–67, 66–67, 66, 67

data trimming 323, 332

debugging 175, 242

decimal format 31

decomposition of sum of squares 360

degrees of freedom 28, 275, 279, 295, 382, 385–86, 385–86, 385–86, 385–86, 390, 399, 442–45, 442–45, 442–45, 442–45, 442–45, 449–53, 449–51, 449–51, 449–56, 449–51, 449–53, 452, 454–56, 454–56, 454–56, 459, 461

Delete A Record 66, 67

delete all the variables on a form 50

Delete Button 38, 80, 270

DESCRIPTIVE Command 249, 299

dichotomous 19, 259, 336, 408, 435–38, 435, 438, 446

dichotomous dependent variable 336

DIF Files 414

DIFFERENCE command 237–38

Disable Skips 57, 67

discriminant function analysis 364

discriminant function coefficients 366

DUMMY command 239–40, 239–40, 239–40, 239–40, 239–40

Dummy variables 69, 239–40, 239–40, 239–40, 239–40, 374

Duplicate A Field 65

Duplicate Button 38, 42

Duplicating Variables 38

## E

ecology 179

ecology option 179

eigenvalue summary table 404

eigenvalues 369

ELSE keyword 227

E-Mail Surveys 71, 416

Entropy 278

Equiweighting 280

exclude missing data 211

execution time 7

exploratory research 15, 17

export 10, 12, 158, 411–14, 411–14, 411–14, 411–15, 411–13

## F

Field Placement 52–53, 52–53, 52–53, 52–53

field width 31–32, 31–32, 31–32, 32, 35, 46, 104, 133, 155, 205, 213–15, 213–14, 213–16, 215–16, 215–16, 215–16, 216, 222, 424

file names 1, 10–12, 10–12, 10–12, 10–12, 10, 29, 92, 160, 173–74, 418

final coding 267–71, 267–71, 267–69, 269–71, 269–71, 270–71

Find Dialog window 58, 64, 178–79, 178–79, 178–79, 178–79

Find Next 58, 64, 178

Find Records 64–65, 64–65, 64–65, 64–65, 64–65

Find Text 65, 178

finite population correction factor 28, 256, 285, 303, 455

Fisher's Exact Test 444–45, 444–45, 444–45, 444–45, 445

fixed factors 380

Fixed format 68, 413, 424

font name 51, 107, 109

font size 51, 107, 109

FOOTNOTE command 204

force variables 333, 371

forcing the constant to zero 324

Format Statement 31–32, 31–32, 31–32, 31, 32, 35–36, 35–36, 35, 41, 419–20, 419–20, 419–20, 419–20, 419–20, 423–24, 419–20, 423–24, 423–24, 423

F-ratio 327, 368, 379

FREQUENCIES Command 252, 266–67, 266–67, 266–67, 266

## G

Gamma 278

global options 179–80, 179–80, 179–80, 179–80, 179–80

goal definition 15

goals and objectives 15

Goodman's interaction analysis 279

goodness-of-fit 321

Graphics 3, 77, 98, 123, 153, 166, 186, 189

Grid 36–41, 36–41, 36–40, 37–41, 37–41, 42, 84, 89, 423

## H

HEADING command 202–4, 202–4, 202–4, 202–4, 203–4

HIDE 128, 178, 263–64, 263–64, 263–64, 263–64, 263–64, 287–88, 287–88, 287–88, 287–88, 287–88

hierarchical tree diagram 359

hypothesis 15, 18, 20–21, 20–21, 20, 24–26, 24–26, 24–26, 20–21, 24–26, 309, 327, 353, 368, 399, 448–50, 448–50, 448–50, 448–50, 448–50, 453–54, 453, 454, 461–62, 461–62, 461–62, 461–62

## I

IF-THEN 191, 210, 226–31, 226–31, 226–28, 226–31, 226–28, 230–31, 230–31, 249, 266, 305, 420

import 1, 41, 69–71, 69, 72, 76–79, 76–79, 78, 153, 163, 411–16, 411–16, 411–16, 411–16, 411–13

increase the field width of a variable 215

Independent Groups 24, 26, 309–10, 309–12, 309–12, 309–12, 310–12, 311–12, 453–54, 453–54, 453–54, 453–54, 453–54

inferential 439, 451

Insert Button 38

Installation 2–4, 2–4, 2–4, 2–4, 2, 89, 92

Interaction Analysis 279

interactive 1, 185, 325, 335, 341, 372

interactive prediction 325, 335, 341

internet 1, 3–4, 3–4, 9–11, 9–11, 9–10, 9–10, 29, 35, 41, 71, 76–118, 76–118, 77–79, 77–79, 84–89, 84–89, 101, 108, 118–20, 118–20, 140, 143–47, 143–47, 147, 150, 163–66, 163–66, 163–65, 163–65, 411, 416, 434

internet surveys 9, 77

Internet Surveys 9, 35, 71, 77–118, 77–118, 84–85, 84–85, 120, 147, 416

inverted matrix 329

iterations 341

## J

join words 269

## K

Kendall's Tau Statistics 278  
keyword BY 274, 281, 306  
Keyword Help 192  
Keyword Index 191  
keywords LO and HI 219  
Kolmogorov-Smirnov 302  
Kruskal-Wallis test 382  
Kurtosis 302

## L

LABELS command 204–6, 204–6, 204–6, 204–5, 205–6, 215, 264, 285–88, 285–88, 285–88, 285–87, 290  
LAG command 236–37, 236–37  
least squares line 322  
leptokurtic 302, 309  
LET command 215–16, 215–16, 215  
liability 8–9, 383  
library 37–38, 37–38, 37, 41  
linear regression equation 318  
LIST command 211, 229, 248–50, 248–50, 248–50, 248, 250, 421, 434  
log of the likelihood function 339  
logistic regression 336  
Lotus 411, 413

## M

Mann-Whitney U 312–13  
manually change records 63  
matrix inversion 329  
maximum number of columns 31  
Mean 19, 24, 27–28, 27–28, 27–28, 27–28, 27–28, 31–32, 31–32, 31–32, 31–32, 32, 34–35, 34, 45, 198, 214–15, 214, 220–21, 220–21, 220–21, 220–21, 221, 224–25, 224–25, 224–25, 224–25, 235, 243–45, 243–45, 243–45, 243–45, 245, 265–66, 265–66, 265, 284–85, 284–85, 284–85, 284–85, 284, 290, 243–45, 290, 297, 300–302, 300–303, 300–302, 300–302, 300–302, 306, 309, 314, 322–23, 322–23, 322–23, 265–66, 326, 335–37, 335–37, 290, 335, 337–38, 338, 340, 343, 348–49, 348–49, 300–302, 290, 355, 361, 363, 365, 372–73, 372–73, 322–23, 372–73, 378, 380, 395–96, 395–96, 395–96, 395–96, 399–401, 399–401, 399–401, 401, 407, 421, 439–42, 439–42, 439–40, 439–42, 439–42, 442, 451–57, 451–57, 451–57, 451–57, 463  
mean absolute percent error 322  
mean percent error 322  
mean squared error 322  
mean substitution 335, 340, 349, 355, 363, 372, 399

Median 27, 215, 300–301, 300–301, 300–302, 300–301, 300–301, 321, 451  
MERGE command 202, 267  
Merge Procedure Files 184  
Merging Data Files 417  
mesokurtic 302  
methods of research 16  
minimum percent 270  
missing data 35, 45, 65–66, 65–66, 65–67, 66–67, 66–67, 67, 211, 219, 222, 224, 228, 248, 253, 264, 211, 264, 274, 286, 296–97, 296, 299, 335, 340, 349, 355, 363, 372–73, 372–73, 373, 431  
Mode selection 185  
Move Variables 38  
multicollinearity 346  
multiple correlation 396  
multiple regression 325  
multiple regression equation 326  
multiple response 22, 30, 32, 38, 42–45, 42–44, 42–45, 42–45, 41–45, 55, 73–74, 73–74, 73, 84–85, 84, 120, 133–34, 133, 136–37, 136–37, 139–40, 139–40, 147, 205, 229, 250–52, 252, 257–59, 257–61, 257–58, 257–58, 257–59, 261, 267, 288–90, 288–90, 288–89, 290, 297, 435–38, 435–38

## N

N Equals option 294  
net category 272  
network 2–4, 2–4, 2–4, 3  
NEW command 213–14, 213–14, 213–14, 213–14, 213, 222–23, 222–23, 222–23, 222–23, 222–23, 225, 285  
No Delay 186  
non-hierarchical cluster analysis 356  
non-missing 211, 224–26, 224–26, 224–26, 224–25, 224–26, 228, 264, 301  
non-parametric 312, 382  
normal distribution 27, 279, 302, 312–13, 312–13, 312–13, 336, 341, 440–41, 440–41, 440–41, 441  
normalize 191–92, 191–92, 191, 235–36, 235–36, 235–36, 235–36  
NORMALIZE command 235–36, 235–36, 235–36, 235–36, 235–36  
Nth record selection 211  
null hypothesis 15, 20–21, 20–21, 20, 25–26, 25–26, 25–26, 25–26, 25–26, 448–50, 448–50, 448–50, 448–50, 448–50, 453–54, 453–54, 453, 454, 461–62, 461–62, 461–62, 461  
number of days between two dates 223, 433

## O

oblique reference structure analysis 350  
oblique rotation 352  
Odds ratio 278–79, 278–79, 278–79, 278, 279

**One Analysis** 176, 193, 260–61, 260–61, 260–61, 260–61, 264, 300  
**One Variable Per Page** 53  
**one-analysis** 260, 265, 300–301, 300–301, 300–301, 300–301, 300  
**open-ended responses** 31, 229, 266–67, 266–67, 267, 270  
**OPTIONS command** 206–7, 206, 247, 285, 303, 335, 349, 354–55, 354–55, 354–55, 370, 400  
**OR operator** 212  
**order of keywords** 193  
**ordinary least squares** 317  
**Outlier** 331  
**outlier adjustment** 323  
**outliers** 323  
**Output selection** 186

## P

**page heading** 93, 97, 176, 194, 202–3, 202–3, 202–3, 296  
**Page View** 53  
**paper orientation** 179  
**parametric** 21, 283–84, 283–84, 284, 312, 382  
**partial correlation matrix** 328  
**passive variables** 402  
**Pause button** 169, 186  
**percentage base** 207, 257, 287, 308  
**perceptual mapping** 400  
**permanently change an option** 206  
**Phi** 277  
**Piping** 77, 118–19, 118–19, 118–19, 118–19, 127  
**platykurtic** 302  
**Poisson Distribution** 439, 444, 446  
**population** 16–17, 16–17, 16–17, 16–18, 16–18, 27–28, 27–28, 27–28, 16–17, 182–83, 182, 183, 256, 285, 290–91, 290–91, 290–91, 302–3, 302, 290, 302, 312, 318, 358, 380, 426–27, 426–27, 426–27, 426–27, 426–27, 426–27, 429–32, 429–32, 439, 442, 450–51, 450–51, 450–51, 450–51, 450–51, 454–55, 454–56, 454–56, 454–55, 454–56, 462  
**posterior probability** 365  
**powered-vector** 349, 350  
**pre-coding** 267–70, 267–70, 267–71, 267–71, 267, 268–71, 271  
**prediction intervals** 323, 332  
**principal components analysis** 342  
**principal factor analysis** 350  
**print a codebook** 39  
**Print a Procedure File** 184  
**prior probabilities** 370  
**probability of a defined range** 440–42, 440–42, 440–41, 440–42  
**probability of a z value** 440  
**probability of an F -ratio** 443  
**probit regression** 336

**procedure** 1–9, 6–8, 6–8, 6–8, 7–8, 9–10, 9–10, 9, 11–12, 9–10, 12, 20, 25, 31, 66, 71, 76, 88–89, 88–89, 89, 171–85, 171–85, 171–85, 171–85, 171–85, 190–91, 190–91, 191–99, 191, 193–205, 193–203, 193–99, 193–211, 201–3, 201–10, 205–10, 205–11, 206–10, 214–16, 214–15, 214–16, 214–16, 216, 220, 223–31, 225–28, 225–28, 225–31, 225–27, 229, 231–32, 231–36, 232, 234, 236, 241–44, 241–44, 193–205, 241, 247–48, 247–48, 247–48, 247–48, 251, 260, 262, 264, 266, 223, 266–67, 267, 273, 223, 273, 285, 288, 290–92, 290–92, 292, 296–98, 296, 201–10, 297, 317, 247–48, 329, 330, 332–33, 332–33, 332–33, 335, 340, 356–57, 356–57, 357, 371, 373, 380–81, 332–33, 380–81, 385, 411–13, 411–13, 411, 413, 416, 420–23, 420–23, 420–23, 423, 428, 431, 435  
**procedure file** 1–9, 6–8, 9–10, 9, 12, 88, 171–73, 171–73, 171–73, 173, 175–79, 175–79, 175–79, 175–84, 183–84, 183–84, 175–79, 194, 196, 199, 202–4, 202, 204–5, 205, 223, 241–43, 241, 242, 247, 273, 428

**Procedure(s) To Run** 185

**Processing time** 7

## Q

**Quartiles** 304  
**Quick Codebook** 36, 41, 423  
**quotation marks** 210–11, 210–11, 210–11, 210–11, 213, 227, 231, 414

## R

**random digit dialing table** 412, 425–27, 427  
**Random error** 19–20, 19–20, 19–20  
**random factors** 380  
**random number table** 412, 425–26, 425–26, 425  
**random sample** 17, 380, 412, 425–27, 427, 455  
**Rao's F-statistic** 398  
**RECODE command** 205, 215, 218–19, 218–19, 218, 264  
**Record Advance** 56, 63, 67  
**Record Number** 63–66, 63–64, 63–64, 63–66, 66, 68, 169, 196, 208–11, 208, 211, 244, 248–49, 248–49, 248, 271, 273  
**redundancy criteria** 398  
**Regression** 26, 236, 239–40, 239–40, 239–40, 239–40, 244, 278, 315, 317–18, 317–18, 317–18, 320–29, 320–29, 320–27, 329, 331–37, 331–37, 331–41, 338–40, 340–41, 341, 343–44, 343–44, 344, 346–47, 346–47, 346–49, 348–49, 348–49, 364, 368, 374, 393, 408, 439, 459–61, 459–60, 459–60, 459–60  
**regression coefficients** 327, 338  
**regression equation** 322  
**regression line** 322

- relational operators 211–12, 211–12, 211–12, 211–12,  
228–31, 230–31, 230–31, 230–31
- Reliability 19, 315, 316, 384
- REM command 242
- Removing 4, 237, 434
- Replace Dialog window 58, 179
- reserved words 243, 311
- residual 322, 331
- Residual Analysis 279
- residual autocorrelation 323
- response categories 21–22, 21–22, 21–22, 21–22, 32,  
108, 256, 263, 266, 270–72, 270–72, 270–72, 272,  
287
- Results Editor 186, 189
- Rich Text Format 3, 11, 42, 51, 186
- robust regression 320
- r-squared 321, 326
- RSUM command 241–42
- rules governing variable names 32
- Run a Procedure File 183, 184
- Run button 184–85, 184–85, 184–85, 184–85
- RUN command 241

## S

- Sample Size for Means 462
- Sample Size for Percents 461
- Sampling methods 17
- SAVE command 194–95, 194, 197–99, 197–99, 197–  
99, 197–99, 205, 209, 215, 227
- Script 2–4, 2–4, 9, 11–12, 11, 77–78, 77–118, 77–118,  
81–82, 85–91, 98, 106–11, 113, 116, 119, 120–28,  
130, 133–34, 136–38, 140–50, 143–50, 143–50,  
155, 150–56, 150–56
- Search Method 65
- seasonality 237–39, 237–39, 237–39, 237–39
- select a specific variable 57
- SELECT command 175, 194, 208–11, 208–11, 208–11,  
208–11, 242, 249–50, 250, 266, 433
- select non-blank records 211
- Send E-Mail 165
- Send E-Mail Invitations 9, 12, 113, 163, 165, 169
- serial number 8
- simple structure analysis 349
- Skewness 302
- Skip codes 30, 34–35, 34, 39, 43, 55, 57, 67
- slope of regression line 322
- small N 299, 327
- Somers' d 278
- SORT command 198, 232–33, 232–33, 433
- Soundex 65
- Space between Columns 54
- STACK command 216–18, 216–18, 216–18, 216–17,  
402
- stacking variables 402

- standard deviation 27–28, 27, 235, 266, 284, 300–302,  
300, 302, 306, 314–15, 314–15, 322, 327, 331,  
363, 378, 396, 439–41, 439–42, 439, 441–42, 442,  
451–55, 451–56, 453–56, 460, 462
- Standard Error 28, 256, 279, 284–85, 284–85, 284–85,  
284–85, 303, 309, 314–15, 314–15, 315, 322, 324,  
327, 331, 335, 338, 348, 451–52, 451, 454–56,  
454–56, 454–56, 454–55, 458–59, 458–59, 458–  
59, 458, 460–61
- standard error of estimate 322
- standard error of the multiple estimate 327
- standardized coefficients 396
- standardized residual 279, 323, 331
- Starting Columns 38
- Starting Page Number 185–86, 185, 186, 190
- Statistics Calculator 412, 426, 439–40, 439–40, 440,  
454
- StatPac.Ini file 1, 5, 11, 13, 40, 46, 136, 139, 151, 169–  
70, 206, 214, 414, 417
- stub 203, 266, 281–86, 281–84, 281–84, 284–87, 286,  
289–90, 289–90, 296–99, 296–98, 296–98, 296–  
98, 304, 308, 313
- STUDY command 173–74, 173–74, 173–74, 184, 196–  
200, 196–99, 196–200, 202
- study design 1, 9–10, 9, 10, 29, 43, 46–47, 46–47, 46–  
47, 49, 57, 55–170, 61, 66, 69, 85, 201, 213–15,  
213–15, 260, 276, 282, 286, 289, 295, 307–8,  
307–8, 343, 365, 371, 381, 411–13, 411–13, 426
- Style files 89
- subfile 198–200, 198–200, 198–200, 209, 231, 234, 244
- subgroup 305–8, 305–7, 305–8
- SUM command 143, 224–26, 224, 226
- supplemental heading 265, 294
- Support 2, 4–5, 4–5, 8, 7–8, 77
- Systematic error 20

## T

- Tab Delimited 12–13, 113, 156–57, 156–57, 189, 411,  
414–15, 414–15, 414
- Table of Contents 189
- Technical support 2, 7–8
- telephone interviews 16
- template 9–10, 9–10, 10, 45–49, 49–50, 49–50, 50, 61–  
62, 61–62, 62, 68, 186
- Test mode 166, 185
- three-way crosstab 274
- TITLE command 203
- tolerance level 339
- total inertia 401
- TOTAL keyword 286, 296–97, 296–97, 296–97
- transform an existing variable 220
- trend 23, 237–39, 237–39, 237–39
- trouble-shoot a procedure 248
- true aggregate 420, 423



t-test 24, 26, 256, 294, 309–12, 309–12, 309–12, 381, 447–49, 447–50, 447–49, 452–54, 452–54, 454, 459, 461  
 TTEST Command 309–10, 309–10, 309–10  
 two-tailed probability 25, 256, 279, 440–42, 440–42, 440–43, 443, 448–49, 448–50, 448–50, 453–54, 453, 454–56, 454–56, 456, 459, 461  
 two-way crosstabs 274  
 types of variables 30

## U

unbalanced 373  
 unweighted means 373  
 Updating 5, 162  
 Utility Programs 411, 423

## V

V Numbers 176, 200  
 valid codes 30, 33, 39, 43–44, 43–44, 44, 55  
 Validity 9–10, 9–10, 18, 68, 143–49, 143–49, 420  
 Value Label HIDE 263, 287  
 Value labels 30, 32–34, 32–34, 32–34, 33–34, 38–39, 38–39, 38, 39, 41, 42–44, 42–44, 53–55, 53–55, 53–55, 54–55, 74, 84–118, 84–118, 108, 120, 128–33, 128–32, 147, 204–6, 204–5, 204–6, 204–6, 215–17, 215–17, 240, 248, 253–55, 253–55, 253–55, 254–55, 257–60, 257–61, 257, 259–61, 259–61, 264, 266, 276, 281–82, 281–82, 281–82, 281–82, 284–89, 284–86, 285–89, 288–89, 295, 298, 305–8, 305–8, 305–8, 305–8, 371, 381, 403, 414, 424  
 Value Labels Indent 54  
 variability 20, 27, 322–23, 322–23, 327, 332, 341, 344, 361, 369, 373, 380, 387, 389, 401, 451, 457, 460–61, 460–61, 460–61, 460–61  
 Variable Detail 36, 39–42, 39–42, 40, 49, 50–51, 50–51, 55–57, 57, 63, 68, 177–78, 177–78, 177, 423  
 Variable Format 30, 36–38, 38, 42, 46, 225, 227, 413  
 Variable Label Indent 53–54, 53–54  
 Variable List 40–42, 40, 42, 49, 50–51, 50–51, 55, 57, 63, 68, 171–72, 171–72, 171–72, 171–72, 175, 177, 171–72, 177, 197–98, 197–98, 197–98, 197–98, 200–202, 200–202, 200–202, 202, 204, 216–18, 216–18, 224–25, 224–26, 232–33, 224–25, 235–37, 235–37, 238, 248, 253, 258, 261, 265, 274, 281, 285, 288–89, 288–89, 297–98, 297, 299–301, 299–301, 299, 301, 305–6, 305, 309–11, 309–11, 309–11, 309–11, 313, 317, 325–26, 333, 336, 342–43, 342–43, 350, 356, 365, 371, 374–76, 374–78, 377, 382–83, 382, 383–90, 384–90, 394  
 Variable List window 42, 55, 68, 172, 177, 171–72, 177, 202  
 variable name 30, 31–32, 31, 38, 42, 44, 57, 76, 84, 166, 182, 203, 213–16, 213–16, 218, 222–23, 222,

225, 238, 244, 252–53, 252–53, 252–53, 252–53, 266, 281, 285, 298, 304, 308, 313, 316, 325, 336, 342, 349, 355, 364, 382, 400  
 Variable Numbers 39, 55, 68, 73, 236, 248, 258, 289, 326, 356, 365, 394, 420  
 Variable Separation 53  
 Variable Text Formatting 50, 52  
 variable type 30, 212  
 Variance 27, 279, 302, 321, 326, 328, 333, 335, 342–43, 342–43, 345–48, 345–48, 351, 353–54, 353–54, 357, 361, 368, 373, 375, 380–82, 380–82, 383, 384–91, 373, 394, 397, 399, 401, 409, 456, 460  
 variance inflation factors 348  
 variance-covariance matrix 328  
 varimax 349, 350, 354  
 Verbatim Blaster 267–69, 267–69, 267–69, 267, 269, 271, 415  
 verbatim response 22, 267  
 view mode 53

## W

warranty 8  
 web site 5, 9–11, 9–11, 77–84, 77–79, 89, 150, 152, 165, 169, 416  
 WEIGHT command 233–35, 233–35, 233–35, 233–35, 290  
 weighted cross-factor rotation 350  
 weighting 179–83, 179–83, 231, 233–34, 233–34, 233–34, 233, 234, 290, 293, 394  
 Wilcoxon test 312  
 Wilk's lambda 399  
 Wilks' lambda 368  
 word processor 3, 5, 11, 41–42, 41–42, 41, 49, 186, 269, 414  
 WRITE command 195, 197–201, 197–201, 209, 216, 235, 415

## Y

Yate's Correction 279, 295, 444  
 Yule's Q and Yules Y 278

## Z

zero values 254  
 zoom factor 179